



# **TRAINING SUPPORT SYSTEM**

## **Software Requirement Specification**

– Hanoi, August 2022 –

**RECORD OF CHANGES**

| Date   | A*<br>M, D | In charge | Change Description                               |
|--------|------------|-----------|--|
| 9/Sep  | A          | HungNV    | Thiết kế code package (I.1 Code Packages)        |
| 9/Sep  | A          | ManhPV    | Thiết kế Database Schema(II.2 a.Database Schema) |
| 9/Sep  | A          | HoaND     | Mô tả table (II.2 b.Table Description)           |
| 10/Sep | M          | HungNV    | Sửa thiết kế code package (I.1 Code Packages)    |
| 14/Sep | M          | HungNV    | Sửa thiết kế code package(I.1 Code Packages)     |
| 17/Sep | A          | HungNV    | Thêm mô tả cho Home Page (II.1                   |
| 17/Sep | A          | HungNV    | Thêm mô tả cho Dashboard                         |
| 17/Sep | A          | HungNV    | Thêm mô tả cho Login (II.3)                      |
| 18/Sep | A          | ManhPV    | Thêm Code design ChangePassword (II.7)           |
| 18/Sep | A          | ManhPV    | Thêm Code design Setting list (II.8)             |
| 18/Sep | A          | ManhPV    | Thêm Code design Setting details (II.9)          |
| 18/Sep | A          | HoaND     | Thêm mô tả cho Password Reset design(II.4)       |
| 18/Sep | A          | HoaND     | Thêm mô tả cho User Register design(II.5)        |
| 18/Sep | A          | HoaND     | Thêm mô tả cho User Profile design(II.6)         |
| 18/Sep | A          | AnhLH     | Thêm mô tả cho Subject Details (II.10)           |
| 18/Sep | A          | AnhLH     | Thêm mô tả cho Web Contact List (II.11)          |
| 18/Sep | A          | AnhLH     | Thêm mô tả cho Web contact details (II.12)       |
| 18/Sep | A          | DuyDC     | Thêm mô tả cho User List (II.7)                  |
| 18/Sep | A          | DuyDC     | Thêm mô tả cho User Details (II.8)               |
| 18/Sep | A          | DuyDC     | Thêm mô tả cho Subject List (II.9)               |
| 25/Sep | M          | AnhLH     | Sửa phần mô tả của Web Contact List (II.11)      |
| 25/Sep | M          | AnhLH     | Sửa phần mô tả của Web Contact Details (II.12)   |
| 11/Oct | A          | ManhPV    | Thêm phần mô tả cho Assignment list (II.19)      |
| 11/Oct | A          | ManhPV    | Thêm phần mô tả cho Assignment detail (II.19)    |
| 11/Oct | A          | ManhPV    | Thêm phần mô tả cho Eval criteria list (II.19)   |
| 9/Oct  | A          | HungNV    | Thêm mô tả cho System Permission (II.28)         |
| 9/Oct  | A          | HungNV    | Thêm mô tả cho User Authorization (II.29)        |
| 9/Oct  | A          | HungNV    | Thêm mô tả cho Eval Criteria Details (II.30)     |

|        |   |        |  |
|--------|---|--------|--|
| 22/Oct | A | HungNV | Thêm mô tả cho Class Eval Criteria List (II.31)    |
| 22/Oct | A | HungNV | Thêm mô tả cho Class Eval Criteria Details (II.32) |
| 23/Oct | A | AnhLH  | Thêm mô tả cho Attendance Tracking (II.38)         |
| 23/Oct | A | AnhLH  | Thêm mô tả cho Attendance details (II.39)          |
| 23/Oct | A | AnhLH  | Thêm mô tả cho Schedule Attendance (II.40)         |
| 23/Oct | A | ManhPV | Thêm mô tả cho New teams(II.33)                    |
| 23/Oct | A | ManhPV | Thêm mô tả cho team detail (II.34)                 |
| 04/10  | A | Hoand  | Thêm mô tả Milestone List(II.16)                   |
| 04/10  | A | Hoand  | Thêm mô tả new Milestone(II.17)                    |
| 04/10  | A | Hoand  | Thêm mô tả cho Milestone Details(II.18)            |
| 23/10  | A | Hoand  | Thêm mô tả cho Schedule List(II.35)                |
| 23/10  | A | Hoand  | Thêm mô tả cho Schedule Details(II.36)             |
| 23/10  | A | Hoand  | Thêm mô tả cho Team List(II.37)                    |
| 9/11   | M | Hoand  | Sửa phần Package descriptions (I.1)                |
| 9/11   | M | AnhLH  | Sửa phần code packages (I.1)                       |
| 9/11   | M | ManhPV | Sửa phần Database Schema (I.2b)                    |

\*A - Added M - Modified D - Deleted

## Table of Contents

|                         |    |
|-------------------------|----|
| I. Overview             | 11 |
| 1. Code Packages        | 11 |
| 2. Database Design      | 12 |
| a. Database Schema      | 12 |
| b. Table Description    | 12 |
| II. Code Designs        | 13 |
| 1. Home Page            | 13 |
| a. Class Diagram        | 13 |
| b. Class Specifications | 13 |
| c. Sequence Diagram(s)  | 13 |
| d. Database Queries     | 14 |
| 2. Dashboard            | 15 |
| a. Class Diagram        | 15 |
| b. Class Specifications | 15 |
| c. Sequence Diagram(s)  | 16 |
| d. Database Queries     | 16 |
| 3. Login                | 16 |
| a. Class Diagram        | 16 |
| b. Class Specifications | 17 |
| c. Sequence Diagram(s)  | 17 |
| d. Database queries     | 17 |
| 4. Password Reset       | 17 |
| a. Class Diagram        | 17 |
| b. Class Specifications | 18 |
| c. Sequence Diagram(s)  | 18 |
| d. Database queries     | 18 |
| 5. User Register        | 18 |
| a. Class Diagram        | 19 |
| b. Class Specifications | 19 |
| c. Sequence Diagram(s)  | 20 |
| d. Database queries     | 20 |
| 6. User Profile         | 20 |

|                         |    |
|-------------------------|----|
| a. Class Diagram        | 20 |
| b. Class Specifications | 20 |
| c. Sequence Diagram(s)  | 21 |
| d. Database queries     | 21 |
| 7. Change Password      | 21 |
| a. Class Diagram        | 22 |
| b. Class Specifications | 22 |
| c. Sequence Diagram(s)  | 23 |
| d. Database Queries     | 23 |
| 8. Setting List         | 23 |
| a. Class Diagram        | 23 |
| b. Class Specifications | 24 |
| c. Sequence Diagram(s)  | 26 |
| d. Database Queries     | 26 |
| 9. Setting Details      | 26 |
| a. Class Diagram        | 26 |
| b. Class Specifications | 26 |
| c. Sequence Diagram(s)  | 28 |
| d. Database Queries     | 28 |
| 10. Subject Details     | 28 |
| a. Class Diagram        | 28 |
| b. Class Specifications | 28 |
| c. Sequence Diagram(s)  | 29 |
| d. Database Queries     | 29 |
| 11. Web Contact List    | 29 |
| a. Class Diagram        | 29 |
| b. Class Specifications | 29 |
| c. Sequence Diagram(s)  | 30 |
| d. Database Queries     | 31 |
| 12. Web Contact Details | 31 |
| a. Class Diagram        | 31 |
| b. Class Specifications | 32 |
| c. Sequence Diagram(s)  | 32 |

|                          |    |
|--------------------------|----|
| d. Database Queries      | 33 |
| 13. User List:           | 33 |
| a. Class Diagram:        | 33 |
| b. Class Specifications  | 34 |
| c. Sequence Diagram(s)   | 35 |
| d. Database queries      | 35 |
| 14. User Details:        | 35 |
| a. Class Diagram:        | 36 |
| b. Class Specifications: | 36 |
| c. Sequence Diagram(s):  | 36 |
| d. Database Queries      | 37 |
| 15. Subject List:        | 37 |
| a. Class Diagram         | 38 |
| b. Class Specifications  | 38 |
| c. Sequence Diagram(s)   | 39 |
| d. Database Queries:     | 39 |
| 16. Milestone List:      | 39 |
| a. Class Diagram         | 40 |
| b. Class Specifications  | 40 |
| c. Sequence Diagram(s)   | 40 |
| d. Database Queries:     | 40 |
| 17. New Milestone:       | 41 |
| a. Class Diagram         | 41 |
| b. Class Specifications  | 41 |
| Class MilestoneDAO:      | 41 |
| c. Sequence Diagram(s)   | 41 |
| d. Database Queries:     | 41 |
| 18. Milestone Details:   | 41 |
| a. Class Diagram         | 42 |
| b. Class Specifications  | 42 |
| c. Sequence Diagram(s)   | 42 |
| d. Database Queries:     | 42 |
| 19. Assignment List      | 42 |

|                            |    |
|----------------------------|----|
| a. Class Diagram           | 43 |
| b. Class Specifications    | 43 |
| c. Sequence Diagram(s)     | 45 |
| d. Database Queries:       | 45 |
| 20. Assignment Details:    | 45 |
| a. Class Diagram           | 45 |
| b. Class Specifications    | 46 |
| c. Sequence Diagram(s)     | 48 |
| d. Database Queries:       | 48 |
| 21. Eval Criteria List:    | 48 |
| a. Class Diagram           | 48 |
| b. Class Specifications    | 49 |
| c. Sequence Diagram(s)     | 50 |
| d. Database Queries:       | 50 |
| 22. Class List:            | 50 |
| a. Class Diagram           | 50 |
| b. Class Specifications    | 51 |
| c. Sequence Diagram(s)     | 52 |
| d. Database Queries:       | 52 |
| 23. Class details:         | 53 |
| a. Class Diagram           | 53 |
| b. Class Specifications    | 53 |
| c. Sequence Diagram(s)     | 53 |
| d. Database Queries:       | 53 |
| 24. Class Setting List:    | 53 |
| a. Class Diagram           | 53 |
| b. Class Specifications    | 53 |
| c. Sequence Diagram(s)     | 54 |
| d. Database Queries:       | 54 |
| 25. Class Setting Details: | 55 |
| a. Class Diagram:          | 55 |
| b. Class Specifications:   | 55 |
| c. Sequence Diagram(s):    | 55 |

|                                  |    |
|----------------------------------|----|
| d. Database Queries:             | 56 |
| 26. Subject Setting List:        | 56 |
| a. Class Diagram:                | 56 |
| b. Class Specifications:         | 57 |
| c. Sequence Diagram(s):          | 58 |
| d. Database Queries:             | 58 |
| 27. Subject Setting Details:     | 58 |
| a. Class Diagram:                | 58 |
| b. Class Specifications:         | 58 |
| c. Sequence Diagram(s):          | 58 |
| d. Database Queries:             | 58 |
| 28. System Permission:           | 58 |
| a. Class Diagram:                | 58 |
| b. Class Specifications:         | 58 |
| c. Sequence Diagram(s):          | 58 |
| d. Database Queries:             | 58 |
| 29. User Authorization:          | 58 |
| a. Class Diagram:                | 58 |
| b. Class Specifications:         | 58 |
| c. Sequence Diagram(s):          | 59 |
| d. Database Queries:             | 59 |
| 30. Eval-Criteria Details:       | 59 |
| a. Class Diagram:                | 59 |
| b. Class Specifications:         | 59 |
| c. Sequence Diagram(s):          | 59 |
| d. Database Queries:             | 59 |
| 31. Class Eval-Criteria List:    | 59 |
| a. Class Diagram:                | 59 |
| b. Class Specifications:         | 59 |
| c. Sequence Diagram(s):          | 59 |
| d. Database Queries:             | 59 |
| 32. Class Eval-Criteria Details: | 59 |
| a. Class Diagram:                | 59 |

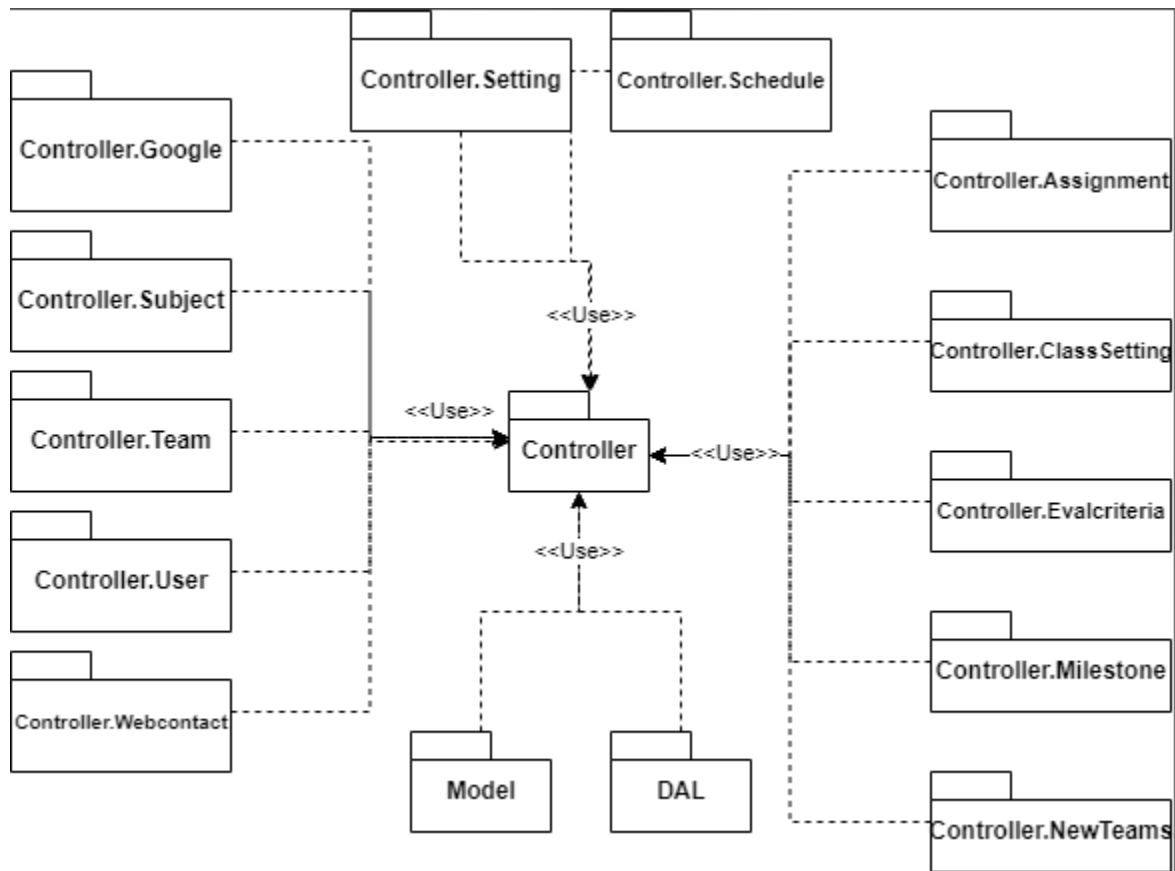


|                          |    |
|--------------------------|----|
| b. Class Specifications: | 59 |
| c. Sequence Diagram(s):  | 59 |
| d. Database Queries:     | 59 |
| 33.New Teams:            | 60 |
| a. Class Diagram:        | 60 |
| b. Class Specifications: | 62 |
| c. Sequence Diagram(s):  | 64 |
| d. Database Queries:     | 64 |
| 34.Team details:         | 65 |
| a. Class Diagram:        | 65 |
| b. Class Specifications: | 66 |
| c. Sequence Diagram(s):  | 68 |
| d. Database Queries:     | 68 |
| 35. Schedule details:    | 69 |
| a. Class Diagram         | 69 |
| b. Class Specifications  | 69 |
| Class scheduleDAO:       | 69 |
| c. Sequence Diagram(s)   | 69 |
| d. Database Queries:     | 69 |
| 36. Schedule List:       | 69 |
| a. Class Diagram         | 69 |
| b. Class Specifications  | 70 |
| Class scheduleDAO:       | 70 |
| c. Sequence Diagram(s)   | 70 |
| d. Database Queries:     | 70 |
| 37. Team List:           | 71 |
| a. Class Diagram         | 71 |
| b. Class Specifications  | 71 |
| Class TeamDAO:           | 71 |
| c. Sequence Diagram(s)   | 71 |
| d. Database Queries:     | 71 |
| 38. Attendance Tracking: | 72 |
| a. Class Diagram         | 72 |

|                          |    |
|--------------------------|----|
| b. Class Specifications  | 72 |
| Class AttendanceDAO:     | 72 |
| c. Sequence Diagram(s)   | 72 |
| d. Database Queries:     | 72 |
| 39. Attendance Details:  | 73 |
| a. Class Diagram         | 73 |
| b. Class Specifications  | 73 |
| Class AttendanceDAO:     | 73 |
| c. Sequence Diagram(s)   | 73 |
| d. Database Queries:     | 74 |
| 40. Schedule Attendance: | 74 |
| a. Class Diagram         | 74 |
| b. Class Specifications  | 74 |
| Class AttendanceDAO:     | 74 |
| c. Sequence Diagram(s)   | 75 |
| d. Database Queries:     | 75 |

## I. Overview

### 1. Code Packages

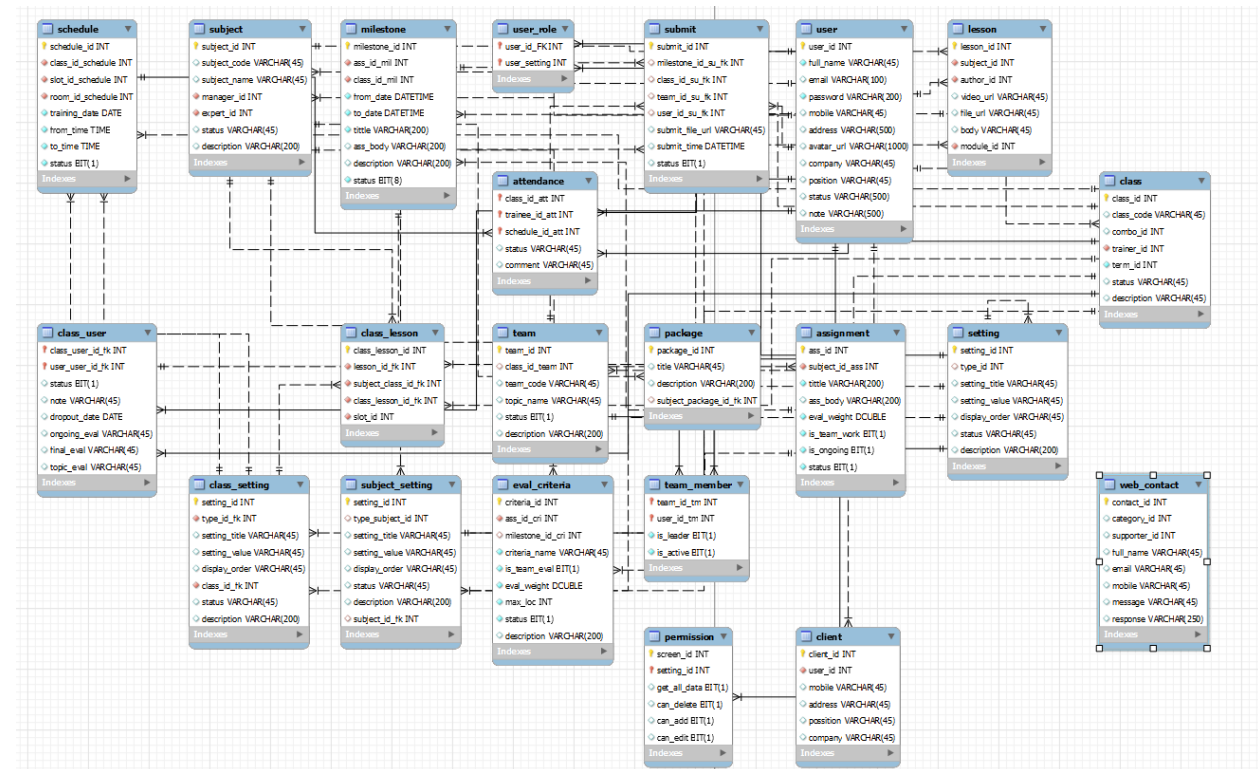


#### Package descriptions

| No | Package    | Description                             |
|----|------------|---|
| 01 | controller | chứa các điều hướng của chương trình    |
| 02 | model      | chứa các thực thể                       |
| 03 | filter     | phân và cấp quyền cho các role          |
| 04 | dal        | kết nối với mysql, lấy data từ database |
| 05 |            |   |
| 06 | ...        |   |

## 2. Database Design

### a. Database Schema



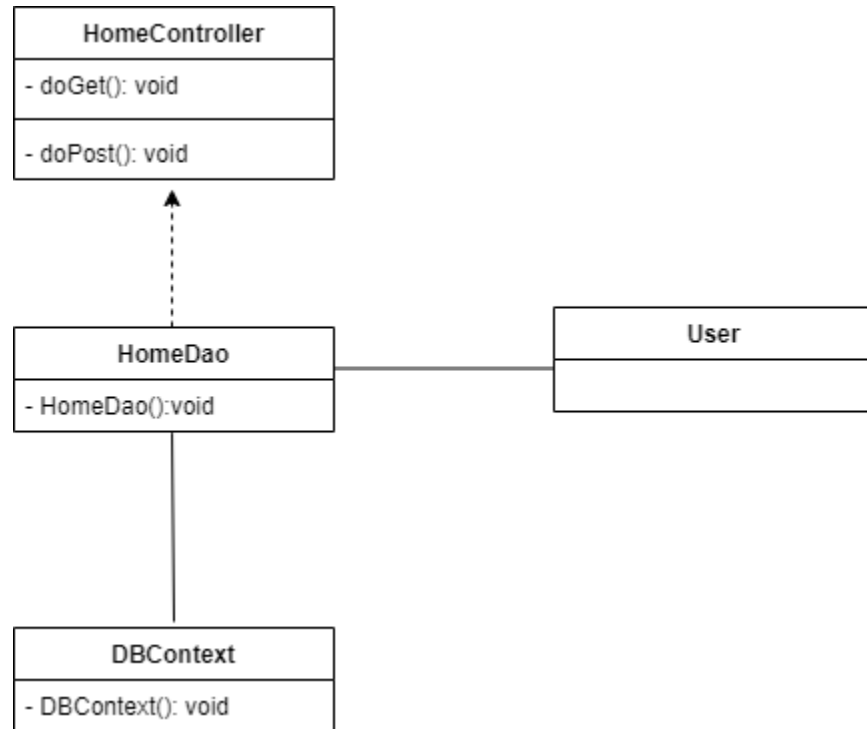
### b. Table Description

| No | Table           | Description  |
|----|-----------------|--|
| 1  | user            | chứa thông tin cơ bản của người dùng như id, password, tên...                        |
| 2  | permission      | bao gồm các quyền của từng người dùng, được chia theo role như thêm, sửa, xóa...     |
| 3  | setting         | lưu thông tin chi tiết của các module  |
| 4  | subject         | chứa các thông tin cơ bản về môn học như mã môn, giáo viên...                        |
| 5  | subject setting | chứa các module của bài học  |
| 6  | lesson          | chứa các thông tin chi tiết của từng bài học theo môn                                |
| 7  | class lesson    | chứa thông tin về lớp học, thời gian học...  |
| 8  | package         |  |
| 9  | client          | chứa các thông tin chi tiết của từng client như địa chỉ, số điện thoại               |
| 10 | class           | chứa thông tin về lớp học như người phụ trách là ai, diễn ra trong kì nào, mã lớp... |
| 11 | class_user      | chứa thông tin của các người dùng có liên quan theo từng lớp                         |
| 12 | class_setting   | lưu thông tin chi tiết trạng thái, thông tin của một lớp học                         |
| 13 | team            | chứa thông tin của các nhóm học sinh trong 1 lớp như mã nhóm, mã lớp, chủ đề...      |
| 14 | web_contact     | chứa các thông tin để liên hệ cho trung tâm  |

## II. Code Designs

### 1. Home Page

#### a. Class Diagram

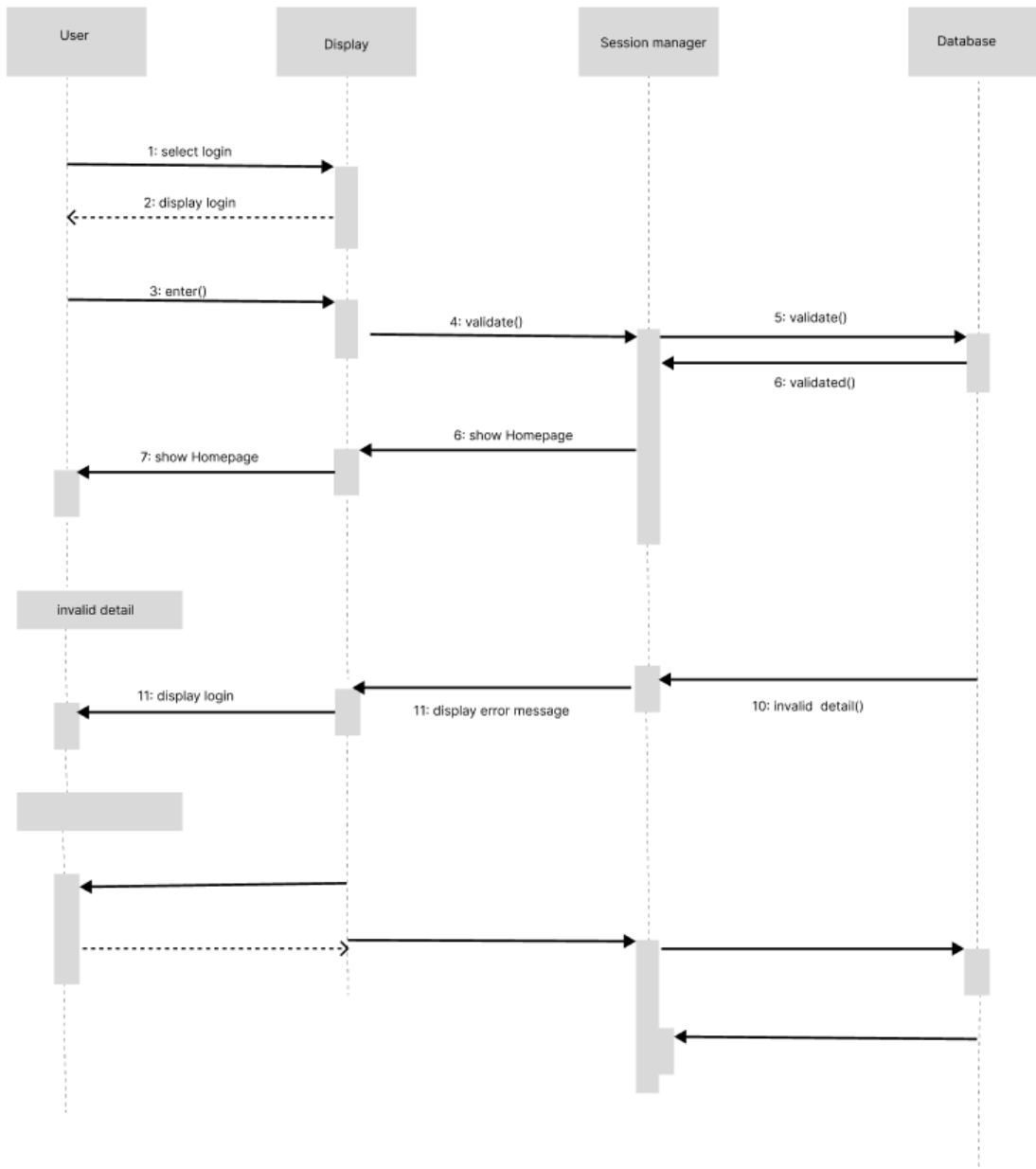


#### b. Class Specifications

##### *LoginController*

| No | Method   | Description         |
|----|----------|---------------------|
| 1  | doGet()  | Handle get request  |
| 2  | doPost() | Handle post request |

#### c. Sequence Diagram(s)



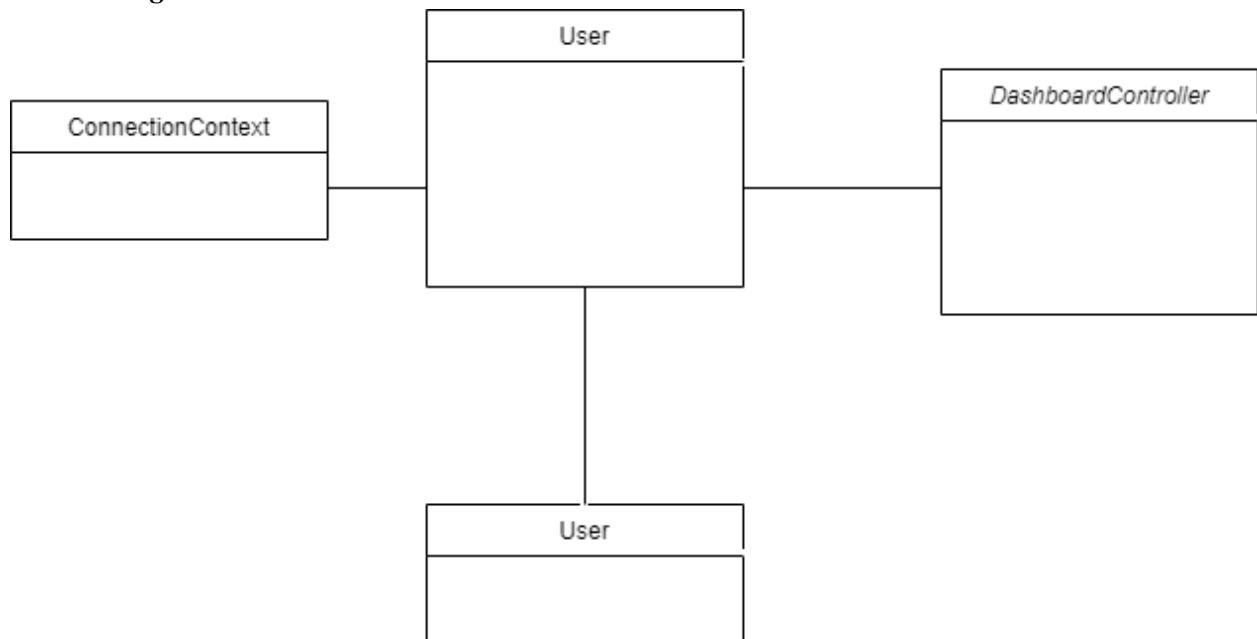
#### ***d. Database Queries***

Query :

“SELECT \* FROM user where email like “+email+” and password like”+password+ ““;”;

## 2. Dashboard

### a. Class Diagram



### b. Class Specifications

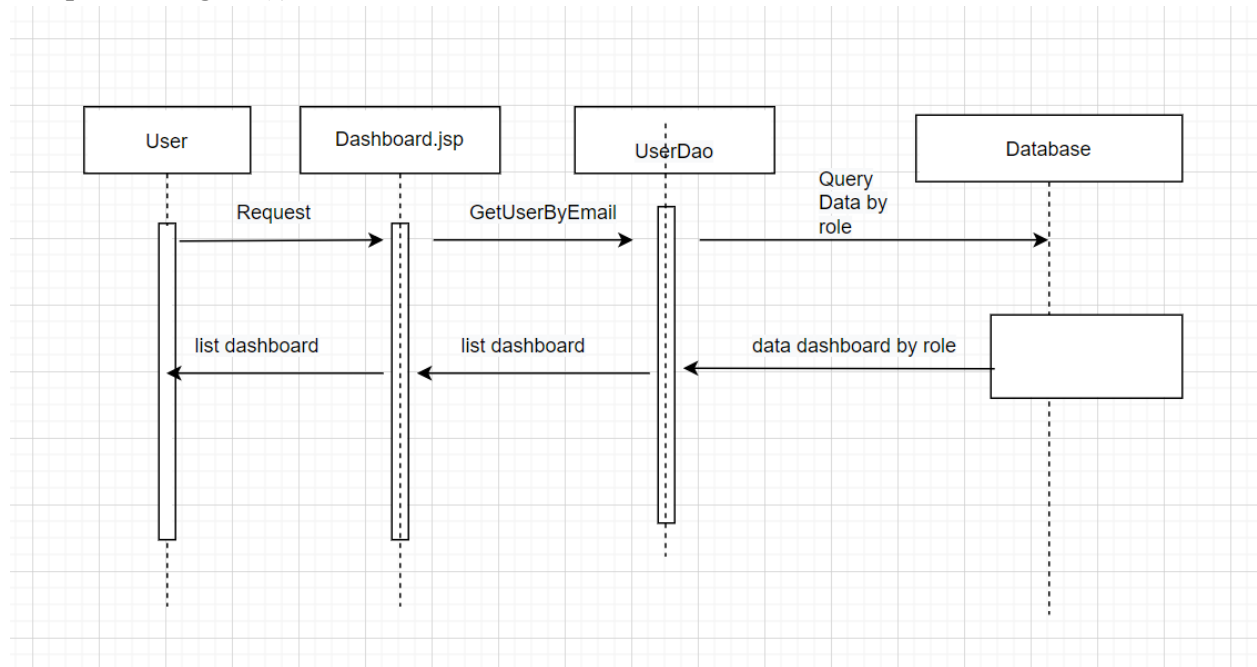
#### DashBoardController

| No | Method   | Description         |
|----|----------|---------------------|
| 1  | doGet()  | Handle get request  |
| 2  | doPost() | Handle post request |

#### UserDao

| No | Method           | Description                    |
|----|------------------|--------------------------------|
| 1  | getAllUser()     | Get all data user              |
| 2  | getUserByEmail() | Get data user which have email |

### c. Sequence Diagram(s)



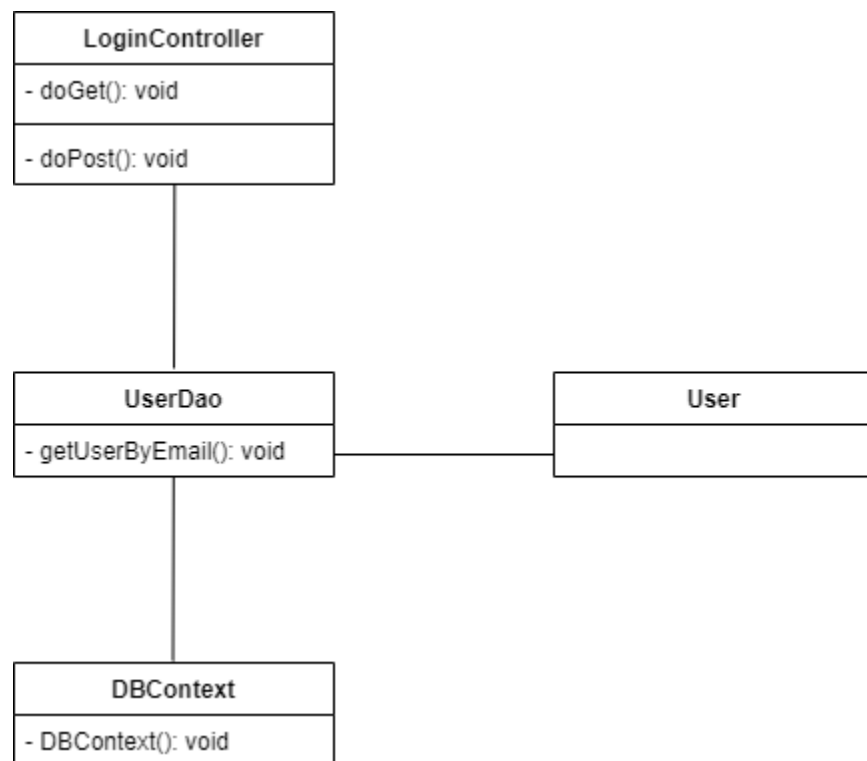
### d. Database Queries

Query :

SELECT \* FROM user where email=?

## 3. Login

### a. Class Diagram





## b. Class Specifications

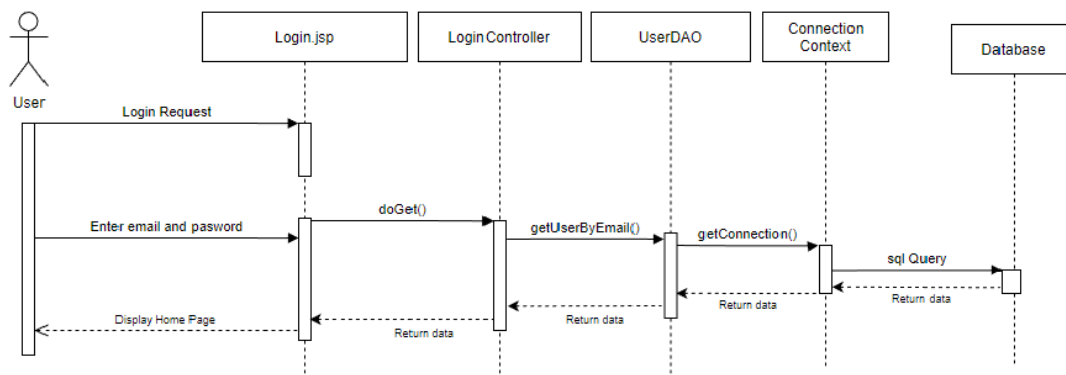
### LoginController

| No | Method   | Description         |
|----|----------|---------------------|
| 1  | doGet()  | Handle get request  |
| 2  | doPost() | Handle post request |

### UserDAO

| No | Method           | Description  |
|----|------------------|--|
| 1  | getUserByEmail() | Get all data from user table which have the identified email |

## c. Sequence Diagram(s)



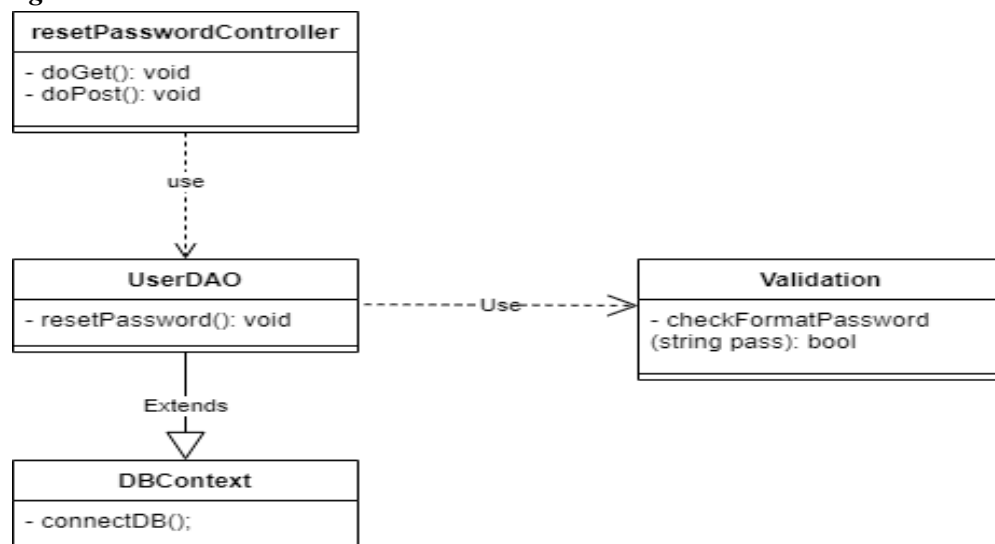
## d. Database queries

Query 1:

“SELECT \* FROM user where email = ? “

## 4. Password Reset

### a. Class Diagram



### *b. Class Specifications*

#### **resetPasswordController**

| No | Method   | Description         |
|----|----------|---------------------|
| 1  | doGet()  | Handle get request  |
| 2  | doPost() | Handle post request |

#### **UserDAO**

| No | Method          | Description                               |
|----|-----------------|---|
| 1  | resetPassword() | send reset password link via user's email |

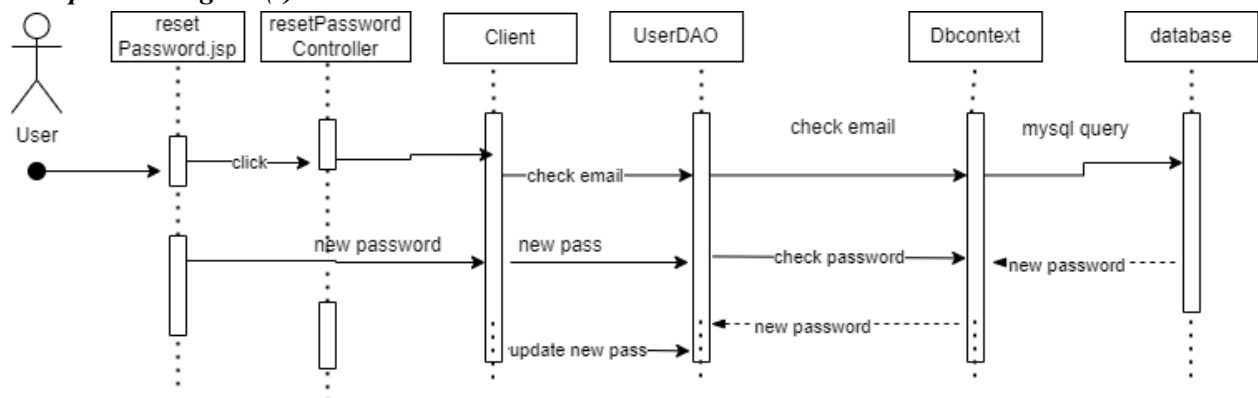
#### **Validation**

| No | Method                | Description   |
|----|-----------------------|---|
| 1  | checkFormatPassword() | Check the new password matches the overriding program pattern |

#### **DbContext**

| No | Method       | Description                 |
|----|--------------|-----------------------------|
| 1  | -connectDB() | Connect program to database |

### *c. Sequence Diagram(s)*



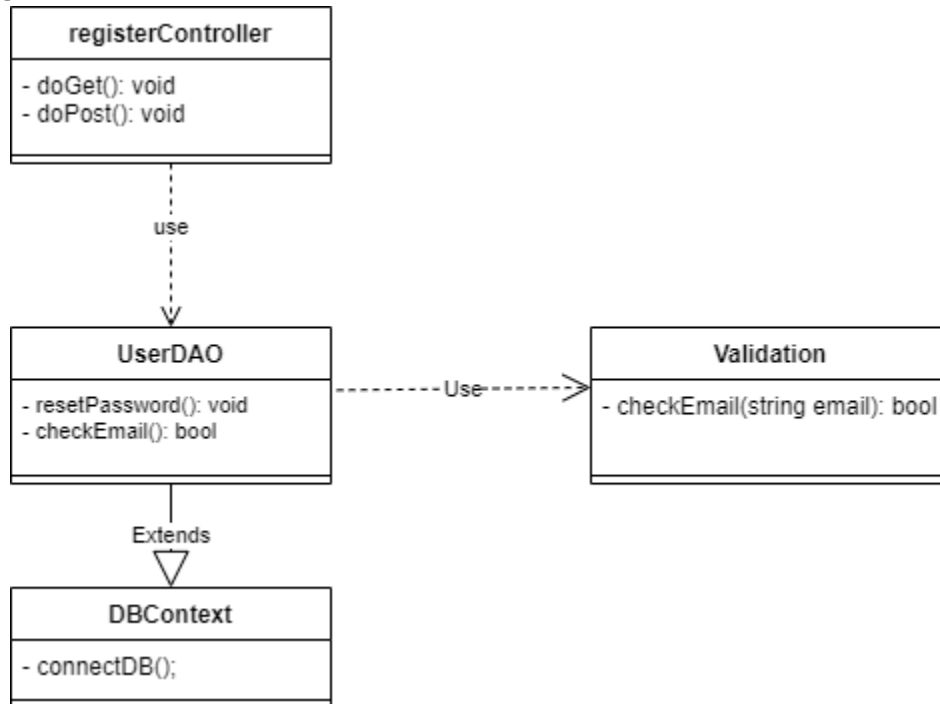
### *d. Database queries*

Query 1:

“SELECT \* FROM user where email = ? “

## 5. User Register

### a. Class Diagram



### b. Class Specifications

#### registerController

| No | Method   | Description         |
|----|----------|---------------------|
| 1  | doGet()  | Handle get request  |
| 2  | doPost() | Handle post request |

#### UserDAO

| No | Method         | Description                        |
|----|----------------|------------------------------------|
| 1  | registerUser() | add new User                       |
| 2  | checkEmail()   | check email is existed in database |

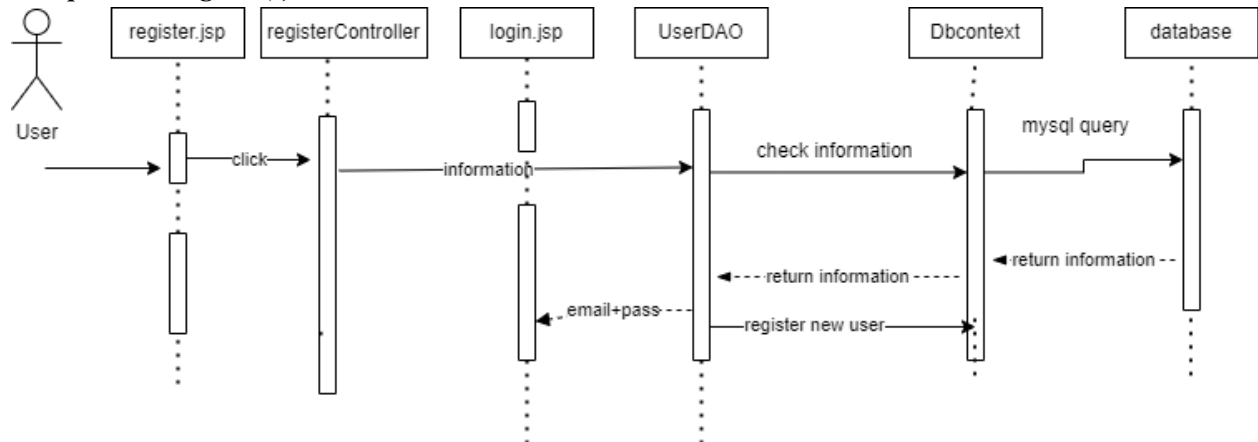
#### Validation

| No | Method                | Description   |
|----|-----------------------|---|
| 1  | checkFormatPassword() | Check the new password matches the overriding program pattern |
| 2  | checkEmail()          | Check email format?   |

#### DbContext

| No | Method       | Description                 |
|----|--------------|-----------------------------|
| 1  | -connectDB() | Connect program to database |

### c. Sequence Diagram(s)



### d. Database queries

Query 1:

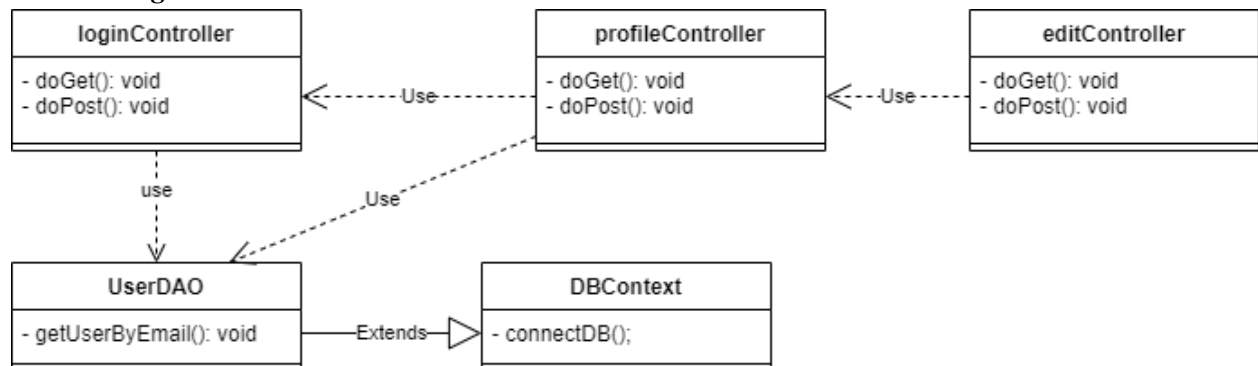
“SELECT \* FROM user where email = ? “

Query 2:

“Update user set password=? where email=? “

## 6. User Profile

### a. Class Diagram



### b. Class Specifications

#### LoginController

| No | Method   | Description         |
|----|----------|---------------------|
| 1  | doGet()  | Handle get request  |
| 2  | doPost() | Handle post request |

#### profileController

| No | Method   | Description         |
|----|----------|---------------------|
| 1  | doGet()  | Handle get request  |
| 2  | doPost() | Handle post request |

#### editController

| No | Method   | Description         |
|----|----------|---------------------|
| 1  | doGet()  | Handle get request  |
| 2  | doPost() | Handle post request |

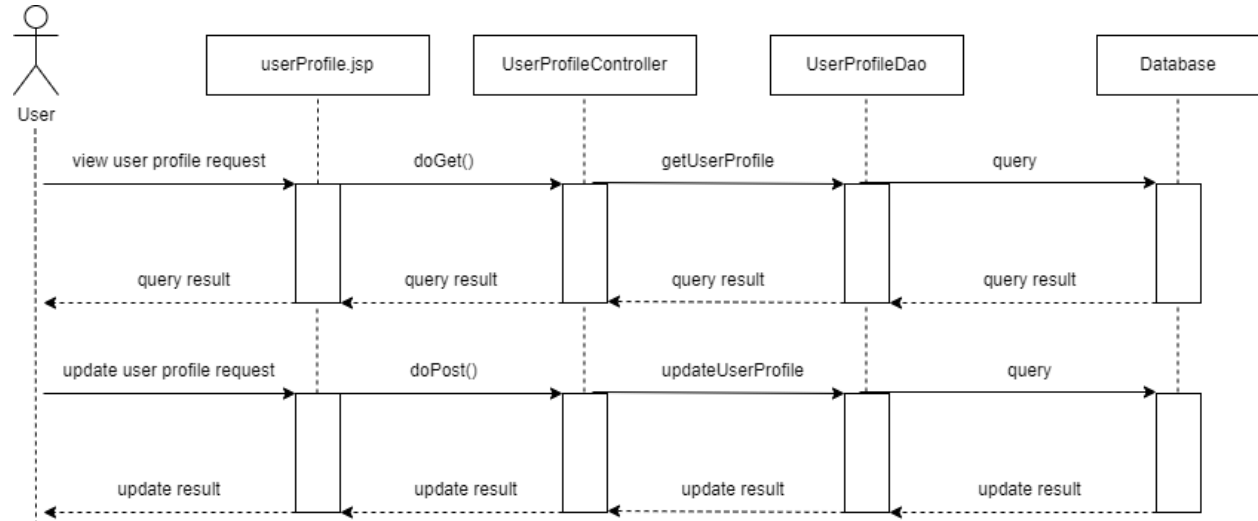
#### UserDAO

| No | Method           | Description  |
|----|------------------|--|
| 1  | getUserByEmail() | Get all data from user table which have the identified email |

#### DbContext

| No | Method       | Description                 |
|----|--------------|-----------------------------|
| 1  | -connectDB() | Connect program to database |

#### c. Sequence Diagram(s)



#### d. Database queries

Query 1:

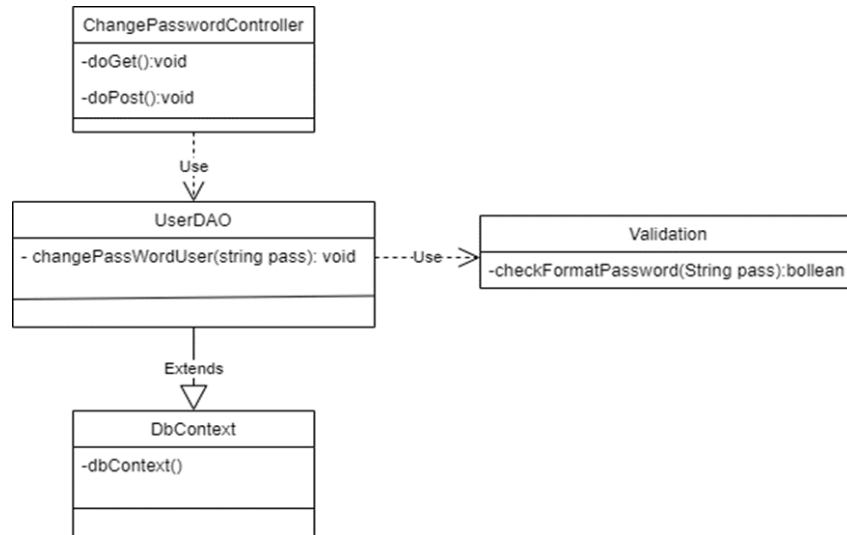
“SELECT \* FROM user where email = ? and password = ?“

Query 2:

“Update user set ? = ? where email=? and password=? “

## 7. Change Password

### a. Class Diagram



### b. Class Specifications

#### *ChangePasswordControler Class*

##### *Class Methods*

| No | Method   | Description   |
|----|----------|---|
| 01 | doGet()  | Get the information(old password ,user id) about user                   |
| 02 | doPost() | Send the new password user want to change to controller<br><br>Redirect |

#### *UserDAO Class*

| No | Method                     | Description                       |
|----|----------------------------|-----------------------------------|
| 01 | changePassWordUser()<br>() | Save password user want to change |

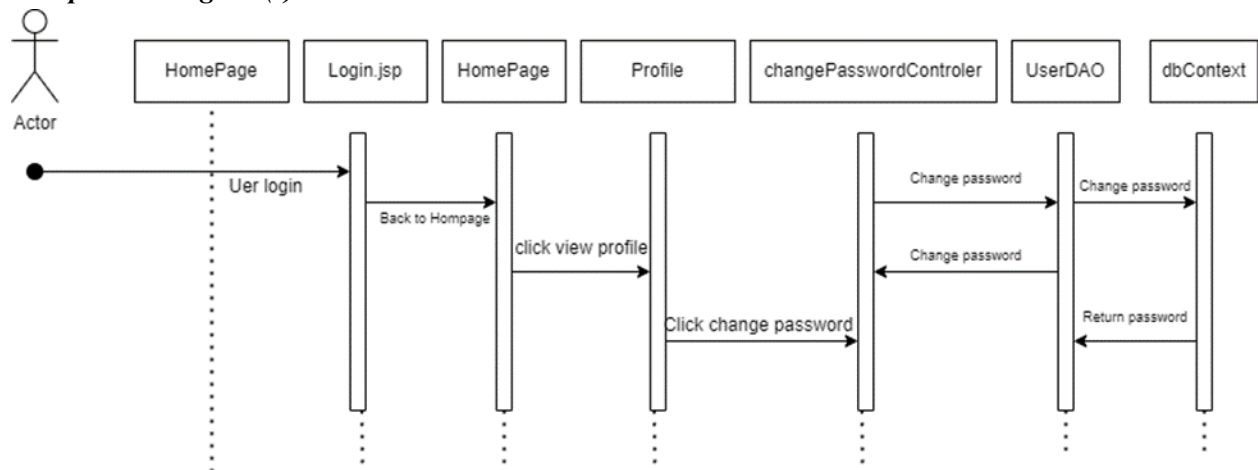
#### *Validation Class*

| No | Method                | Description   |
|----|-----------------------|---|
| 01 | checkFormatPassword() | Check the new password matches the overriding program pattern |

#### *DbContext Class*

| No | Method             | Description                 |
|----|--------------------|-----------------------------|
| 01 | -dbContext()<br>() | Connect program to database |

#### *c. Sequence Diagram(s)*

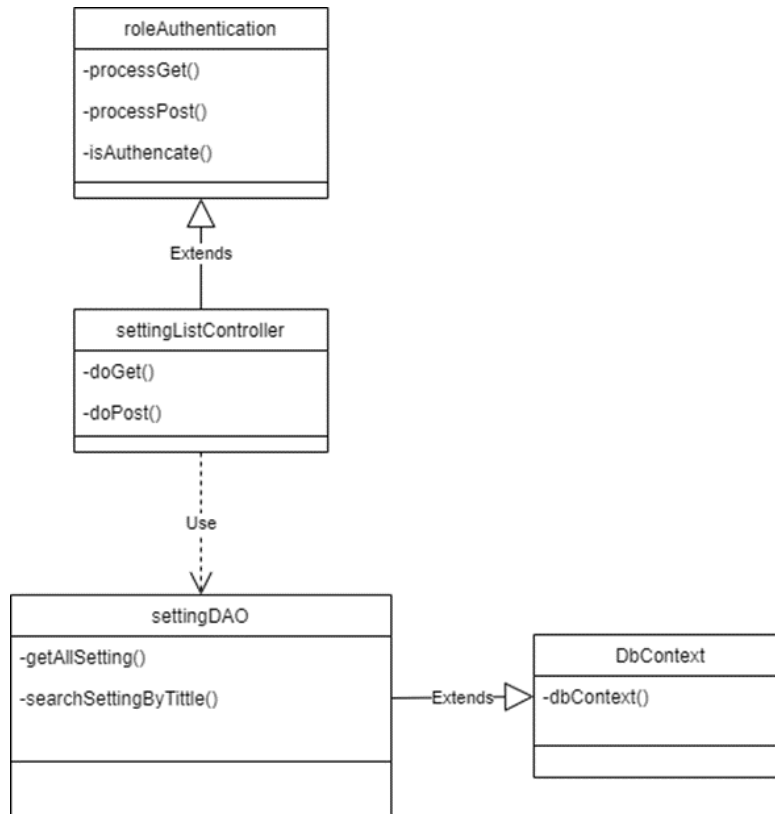


#### *d. Database Queries*

**UPDATE** User **SET** Password= ? **WHERE** ID = 6;

## 8. Setting List

### a. Class Diagram



### b. Class Specifications

#### roleAuthentication Class

| No | Method       | Description   |
|----|--------------|---|
| 01 | isAuthencate | Get the role of the account in session and check if the role matches or not |

#### settingListController Class

#### Class Methods



| No | Method          | Description  |
|----|-----------------|--|
| 01 | <i>doGet()</i>  | <i>Redirect</i>  |
| 02 | <i>doPost()</i> | <i>Send the all settingList from database to frontend</i><br><br><i>Redirect</i> |

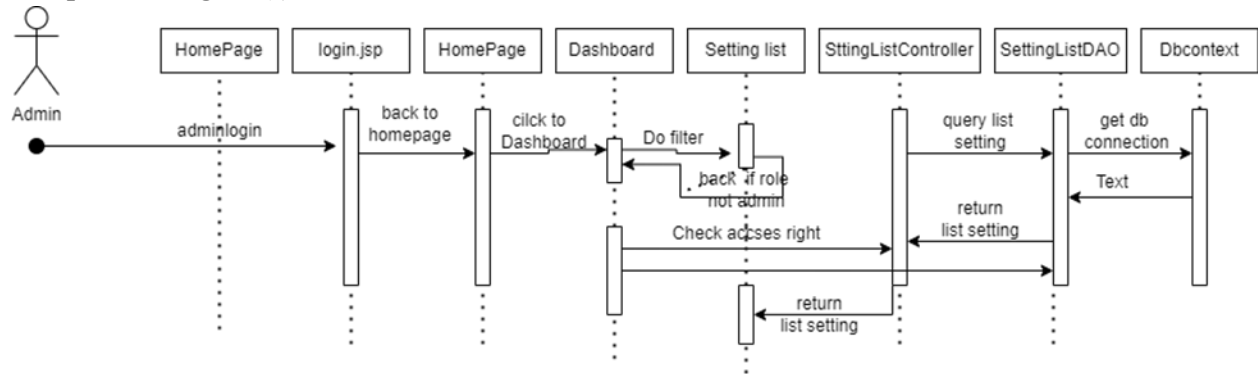
#### *settingDAOClass*

| No | Method                 | Description  |
|----|------------------------|--|
| 01 | getAllSetting()        | Get all data about setting from database               |
| 02 | updateSettingByID()    | Update setting in database                             |
| 03 | searchSettingByTitle() | Find and return setting have title the same user input |

#### *DbContext Class*

| No | Method       | Description                 |
|----|--------------|-----------------------------|
| 01 | -dbContext() | Connect program to database |

### c. Sequence Diagram(s)

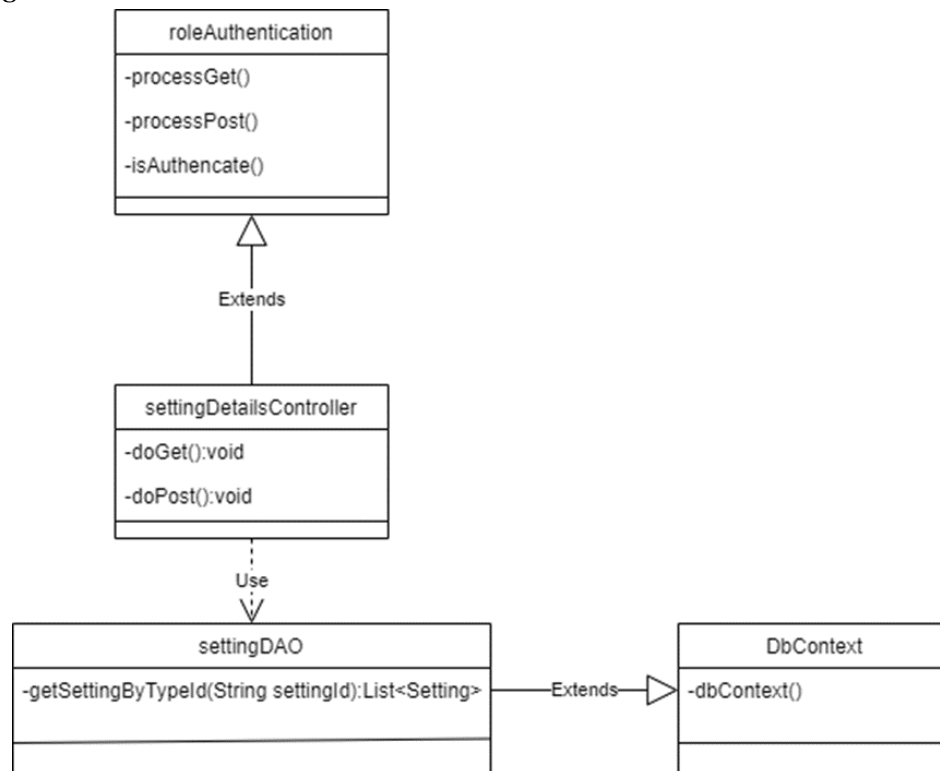


### d. Database Queries

Select \* from settinglist

### 9. Setting Details

#### a. Class Diagram



### b. Class Specifications

#### roleAuthentication Class

| <i>No</i> | <i>Method</i>       | <i>Description</i>   |
|-----------|---------------------|--|
| <i>01</i> | <i>isAuthencate</i> | <i>Get the role of the account in session and check if the role matches or not</i> |

### *settingDeatilsControler Class*

#### *Class Methods*

| <i>No</i> | <i>Method</i>   | <i>Description</i>  |
|-----------|-----------------|---|
| <i>01</i> | <i>doGet()</i>  | <i>Redirect</i>   |
| <i>02</i> | <i>doPost()</i> | <i>Send the all settingList have the same settingId from database to frontend</i><br><br><i>Redirec</i> |

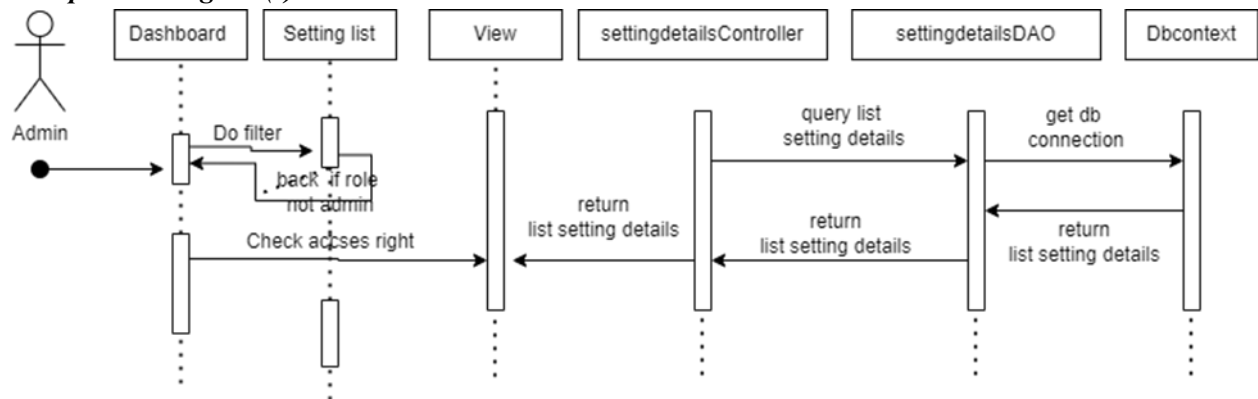
### *settingtDAOClass*

| <i>No</i> | <i>Method</i>               | <i>Description</i>   |
|-----------|-----------------------------|--|
| <i>01</i> | <i>getSettingByTypeId()</i> | <i>Get all settingList have the same settingId from database setting from database</i> |
|           |                             |  |

### *DbContext Class*

| No | Method           | Description                |
|----|------------------|----------------------------|
| 01 | -dbContext<br>() | Connect program to databse |

### c. Sequence Diagram(s)

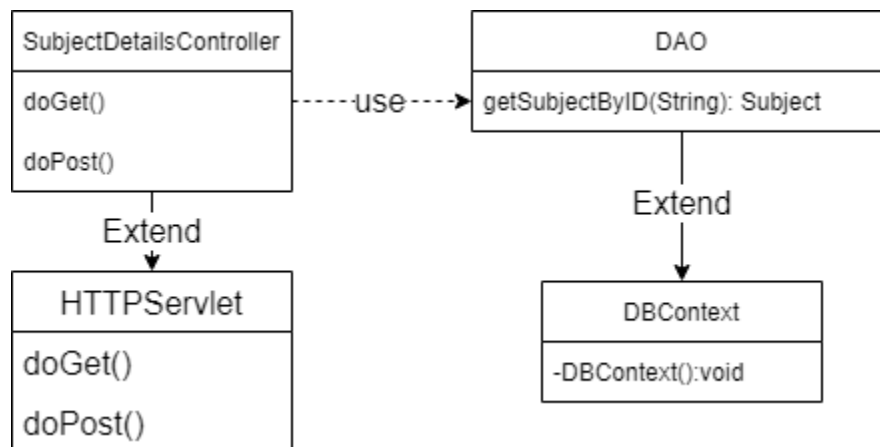


### d. Database Queries

Select \* from settinglist where setting\_id = ?

## 10. Subject Details

### a. Class Diagram

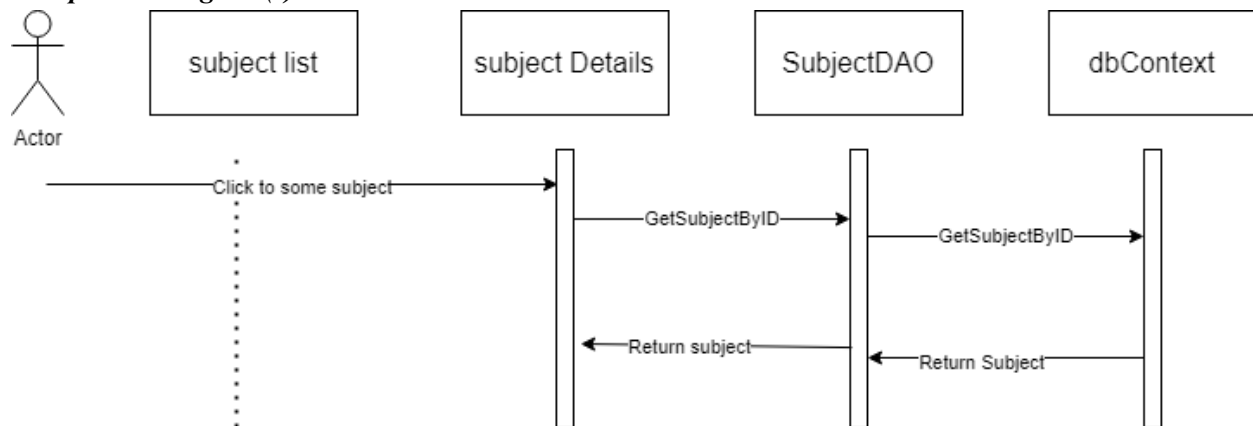


### b. Class Specifications

Class SubjectDAO:

| No | Method         | Description                                   |
|----|----------------|---|
| 1  | getSubjectById | Get a subject have subject_ID equals to input |

### c. Sequence Diagram(s)

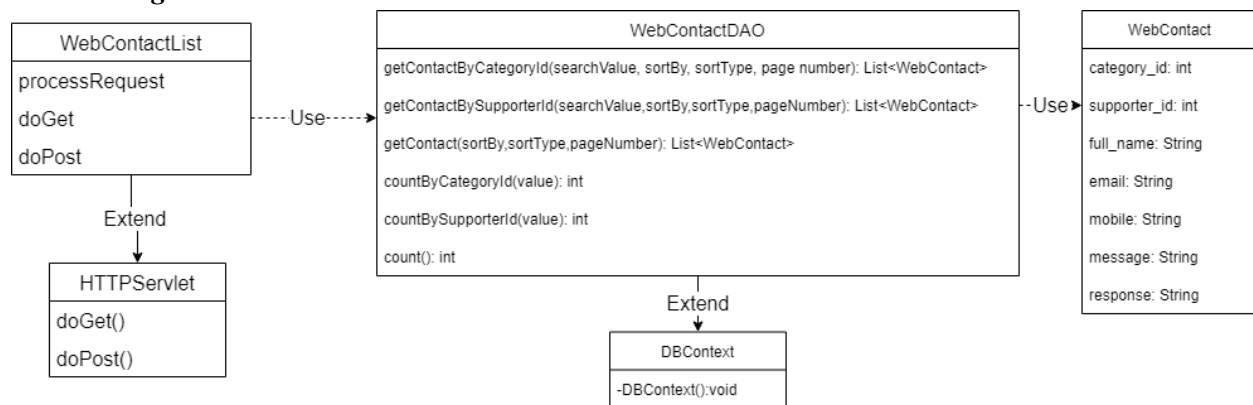


### d. Database Queries

Select \* from subject where subject\_id= ?

## 11. Web Contact List

### a. Class Diagram



### b. Class Specifications

Class WebContactListController:

| No | Method | Description |
|----|--------|-------------|
|----|--------|-------------|

|   |          |   |
|---|----------|---|
| 1 | doGet()  | Process and send data to webcontactlist.jsp |
| 2 | doPost() |   |

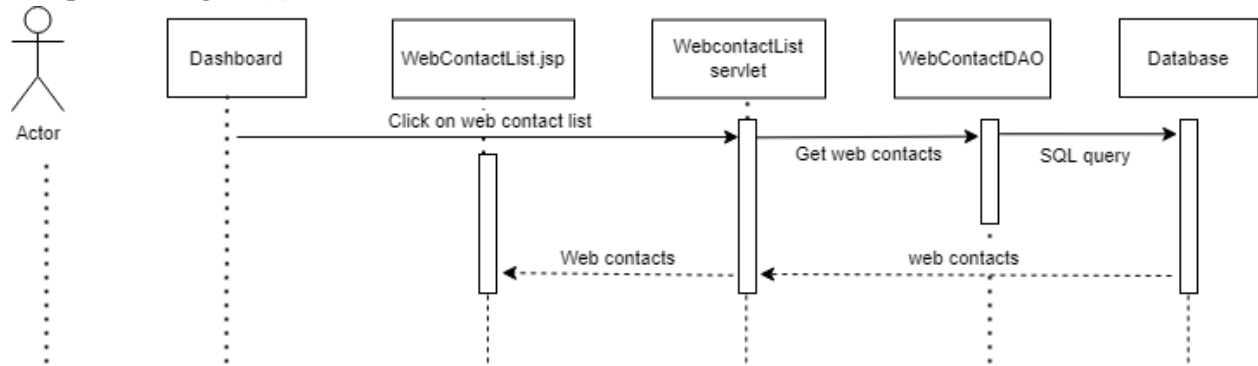
Class WebContactListDAO:

| No | Method                    | Description   |
|----|---------------------------|---|
| 1  | getContactByCategoryId()  | Get contacts have categoryId= searchValue<br>And sort them by option of user                    |
| 2  | getContactBySupporterId() | Get contacts have supporterId= searchValue<br>And sort them by option of user                   |
| 3  | getContact()              | Get contacts and sort them by option of user  |
| 4  | countByCategoryId()       | Count the number of contact have category_id =<br>input value and return maximum of page number |
| 5  | countBySupporterId()      | Count the number of contact have supporter_id=<br>input value and return maximum of page number |
| 6  | count()                   | Count the number of contact and return maximum<br>of page number                                |

Class DBContext: Connect to database

Class WebContact: Contains the information webcontact

### c. Sequence Diagram(s)



### d. Database Queries

SELECT \* FROM web\_contact WHERE category\_id=? ORDER BY category\_id desc LIMIT 5 OFFSET ?

SELECT \* FROM web\_contact WHERE supporter\_id=? ORDER BY category\_id desc LIMIT 5 OFFSET ?

SELECT \* FROM web\_contact ORDER BY category\_id desc LIMIT 5 OFFSET ?

select count(\*) as numberRecord from web\_contact where category\_id=?

select count(\*) as numberRecord from web\_contact where supporter\_id=?

select count(\*) as numberRecord from web\_contact

## 12. Web Contact Details

### a. Class Diagram



### ***b. Class Specifications***

Class WebContactDetailsController:

| No | Method   | Description   |
|----|----------|---|
| 1  | doGet()  | Process, send data to webcontactdetails.jsp               |
| 2  | doPost() | Update web contact, send message to webcontactdetails.jsp |

Class WebContactListDAO:

| No | Method           | Description                                     |
|----|------------------|---|
| 1  | getContactById() | Get and return the contact have id= input value |
| 2  | updateContact()  | update web contact                              |

Class settingDAO:

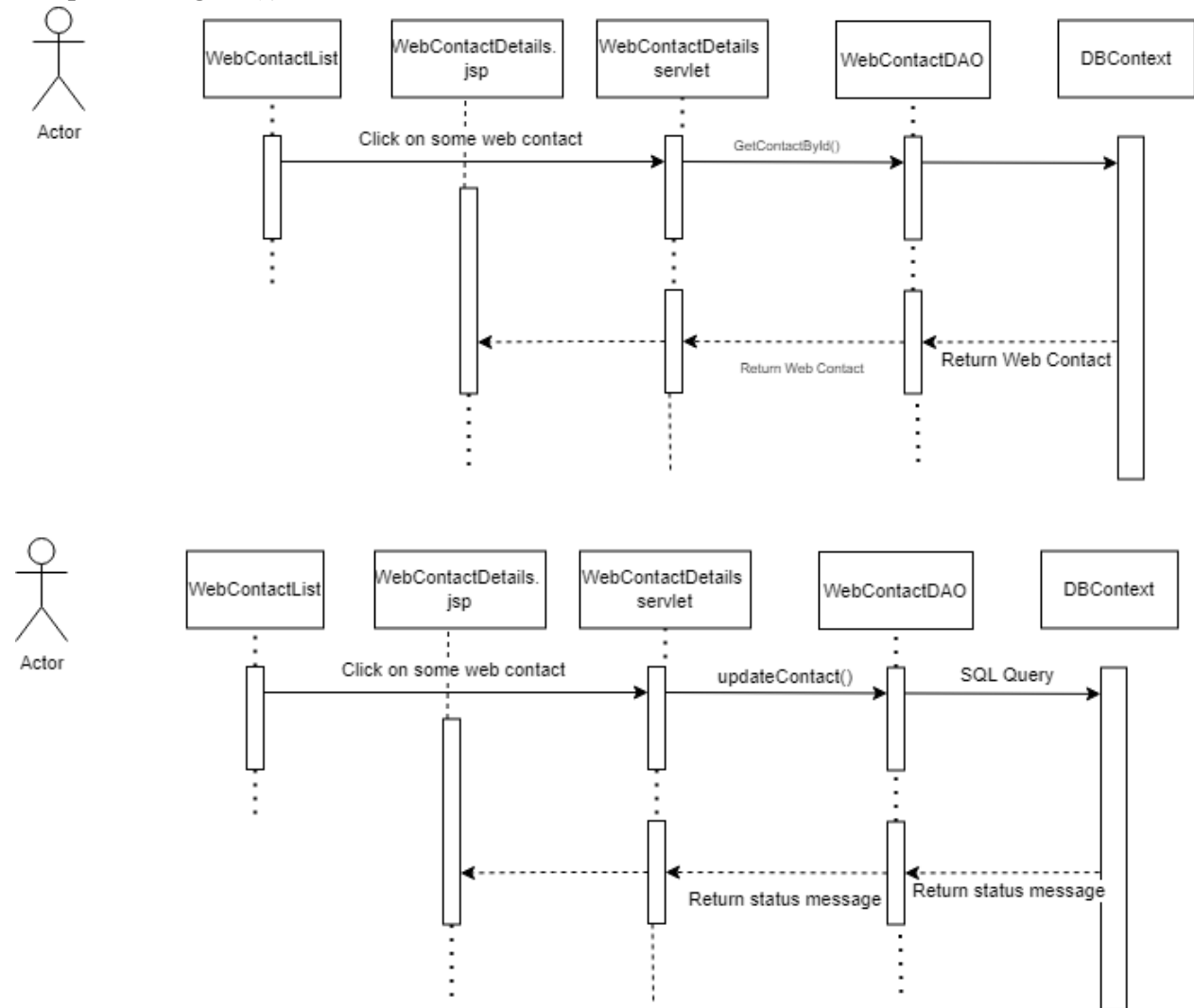
| No | Method           | Description                               |
|----|------------------|---|
| 1  | getCategoryIds() | get all category_id in database           |
| 2  | getSuppoterIds() | get all id of user have role is supporter |

Class DBContext: Connect to database

Class WebContact: Contains the information webcontact



### c. Sequence Diagram(s)



### d. Database Queries

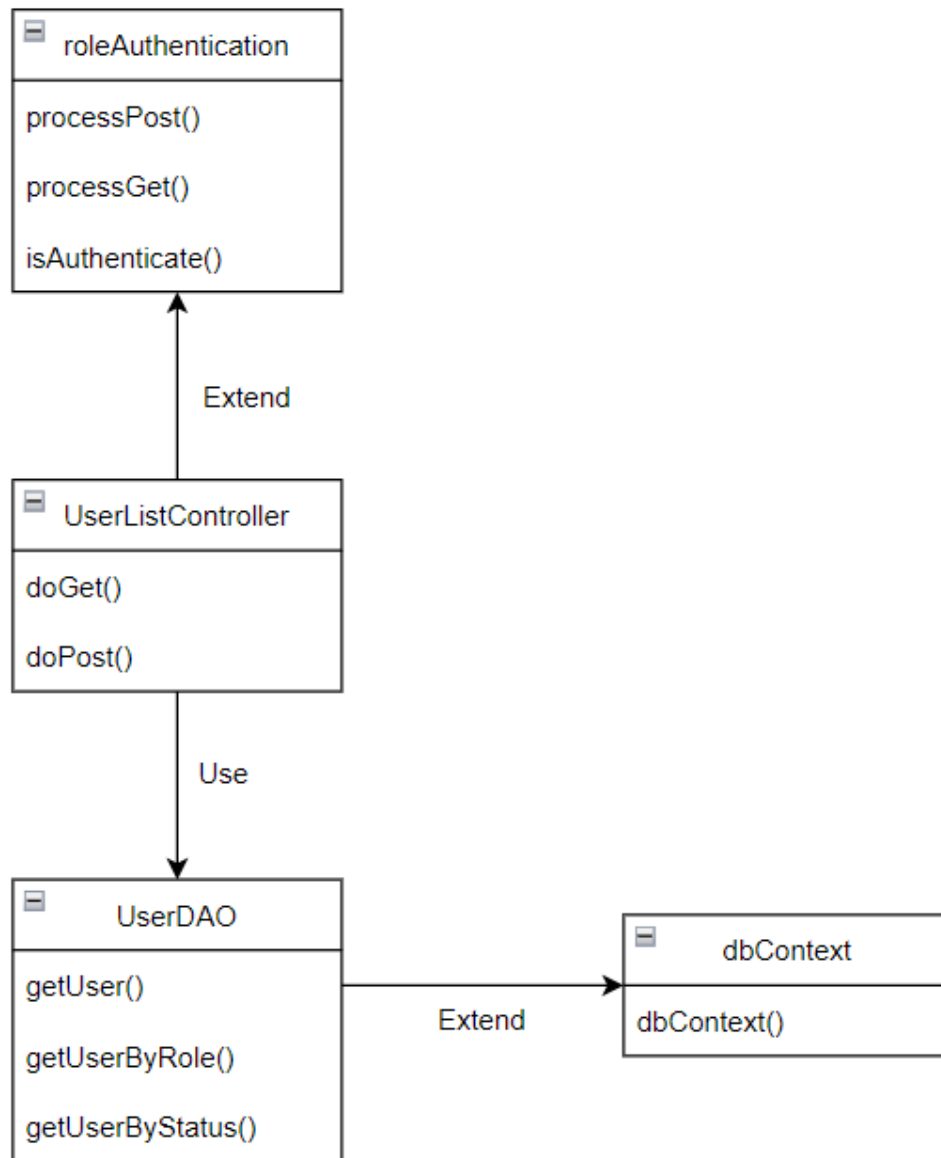
update web\_contact set category\_id=?, supporter\_id=?, response=? where contact\_id=?

select \* from web\_contact where contact\_id=?

select \* from setting where type\_id=2

### 13. User List:

#### a. Class Diagram:



#### b. Class Specifications

Class UserListController

| No | Method  | Description                               |
|----|---------|---|
| 1  | doGet() | Get the information of User for front-end |

|   |          |             |
|---|----------|-------------|
| 2 | doPost() | Edit a User |
|---|----------|-------------|

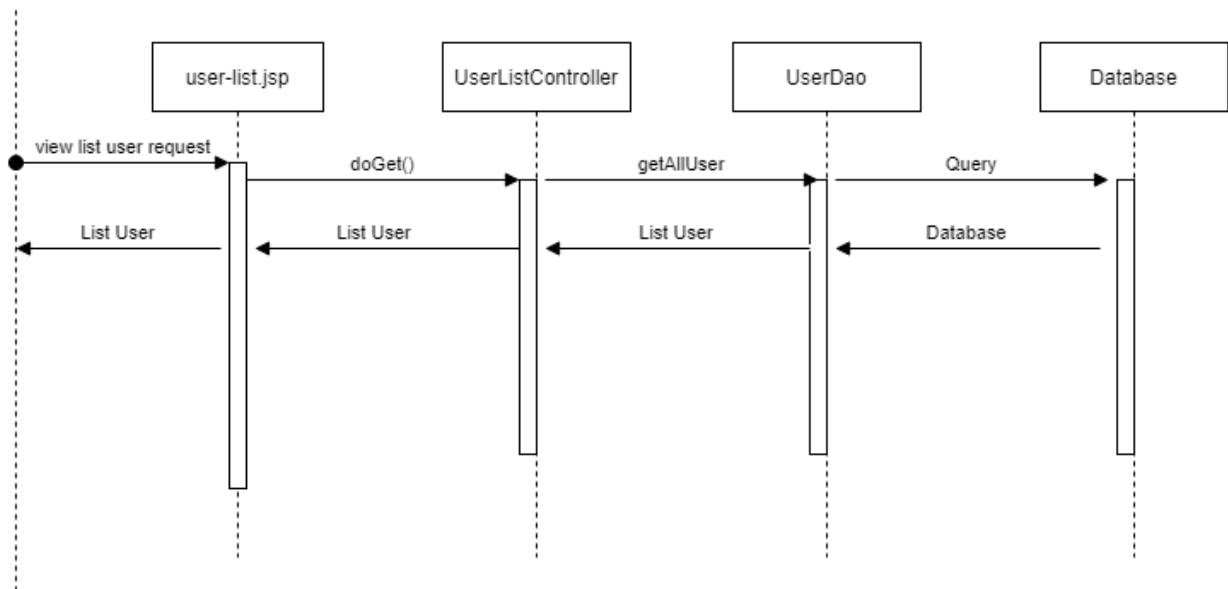
Class UserDao:

| No | Method            | Description               |
|----|-------------------|---------------------------|
| 1  | getUser()         | Get all Users.            |
| 2  | getUserByType()   | Get all User by this Type |
| 3  | getUserByStatus() | Get all User by Status    |

Class DBContext: Connect to database

### c. Sequence Diagram(s)

Admin



### d. Database queries

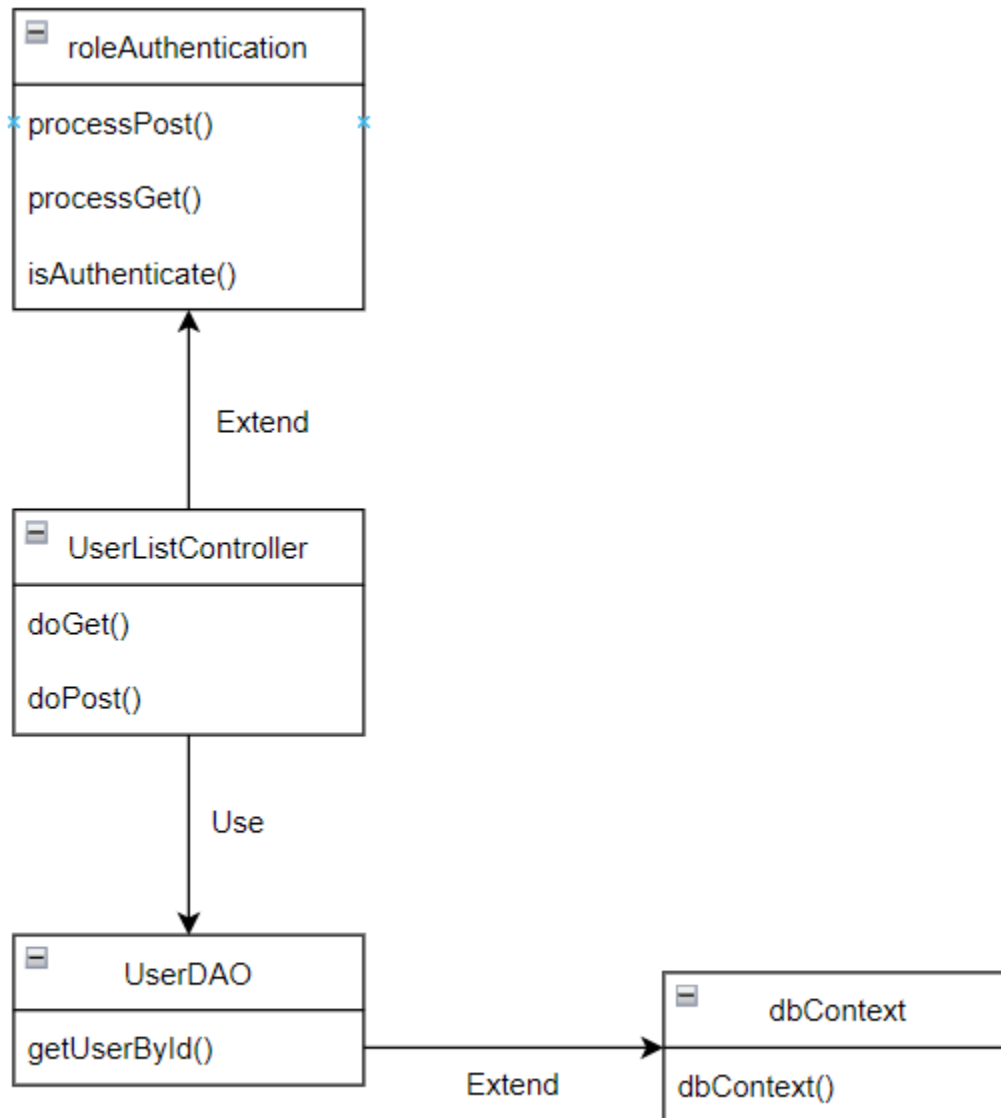
Select \* from User

Select \* from User where user\_role = ?

Select \* from where user\_status = ?

#### 14. User Details:

##### a. Class Diagram:

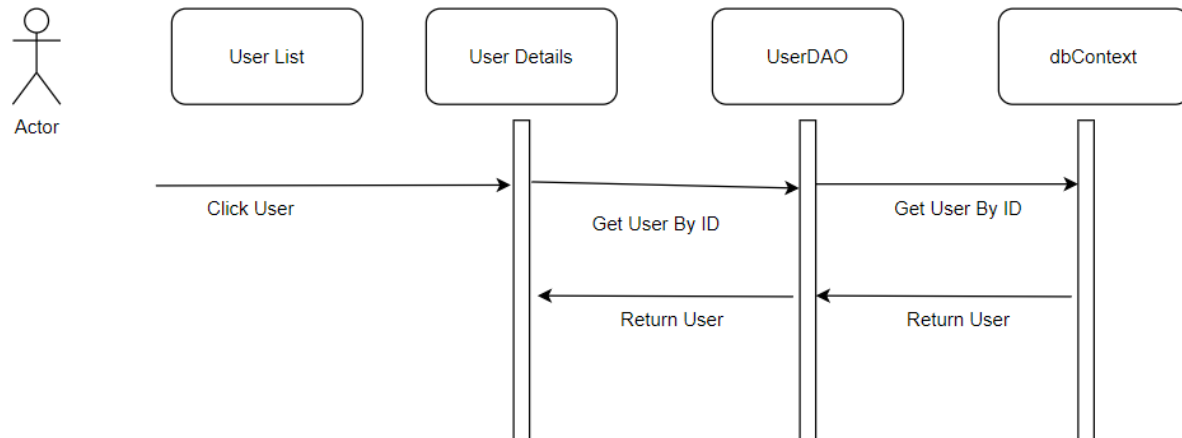


##### b. Class Specifications:

Class UserDAO:

| No | Method      | Description                        |
|----|-------------|------------------------------------|
| 1  | getUserById | Get a User have ID equals to input |

***c. Sequence Diagram(s):***

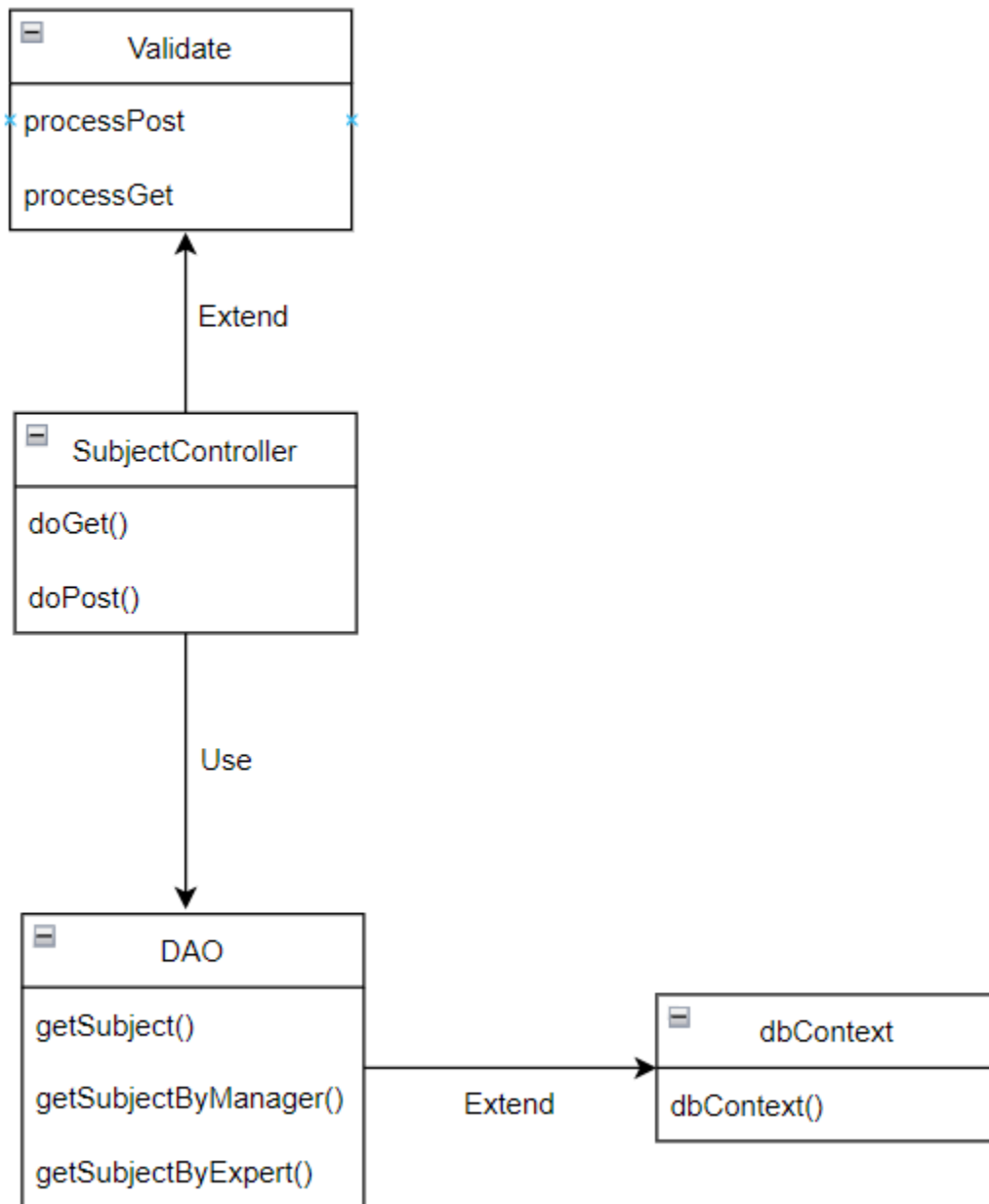


***d. Database Queries***

Select \* from User where user\_id = ?

## 15. Subject List:

### a. Class Diagram



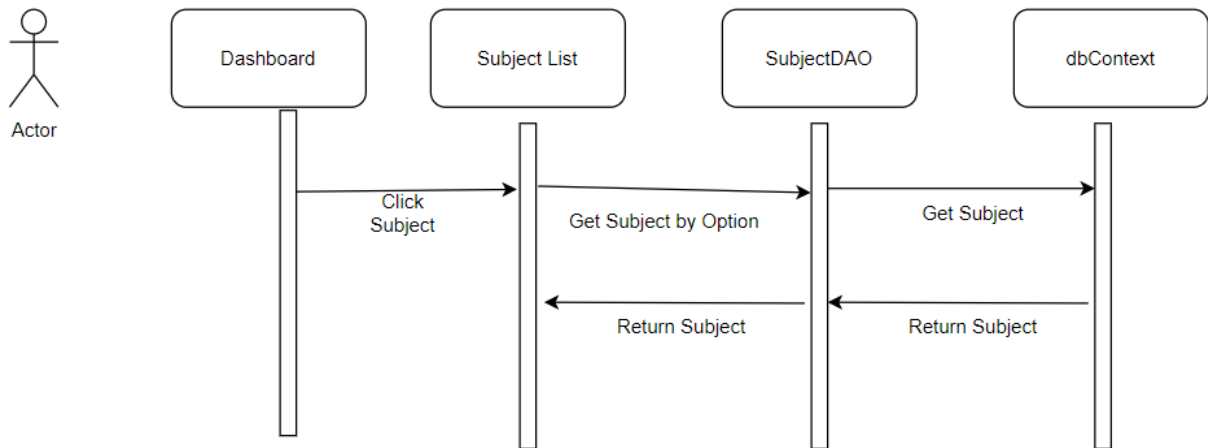
### b. Class Specifications

Class SubjectDAO:

| No | Method | Description |
|----|--------|-------------|
|----|--------|-------------|

|   |                     |                        |
|---|---------------------|------------------------|
| 1 | getSubject          | Get all subject        |
| 2 | getSubjectByManager | Get subject by manager |
| 3 | getSubjectByExpert  | Get subject by expert  |

**c. Sequence Diagram(s)**



**d. Database Queries:**

Select \* from Subject

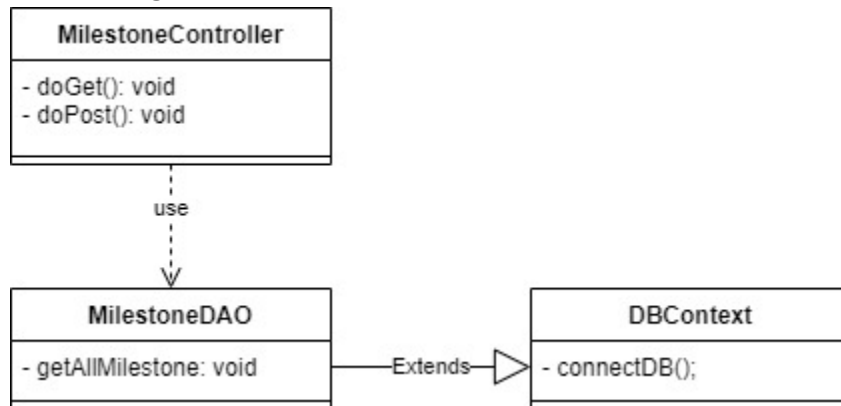
Select \* from Subject where manager\_id = ?

Select \* from Subject where expert\_id = ?

Select \* from Subject where status = ?

## 16. Milestone List:

### a. Class Diagram

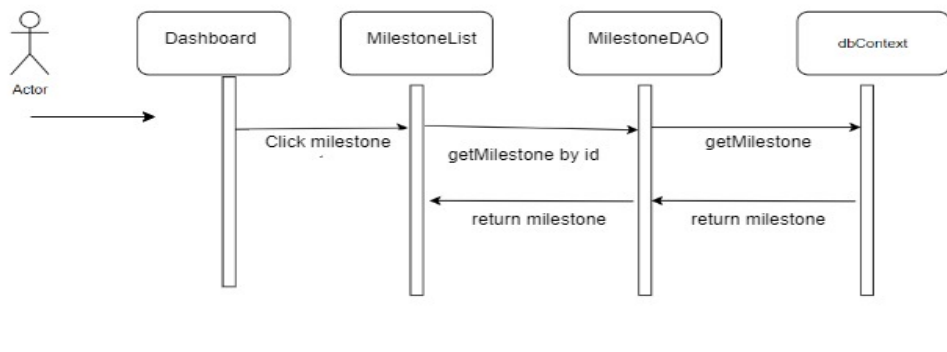


### b. Class Specifications

Class MilestoneDAO:

| No | Method          | Description       |
|----|-----------------|-------------------|
| 1  | getAllMilestone | Get all milestone |

### c. Sequence Diagram(s)



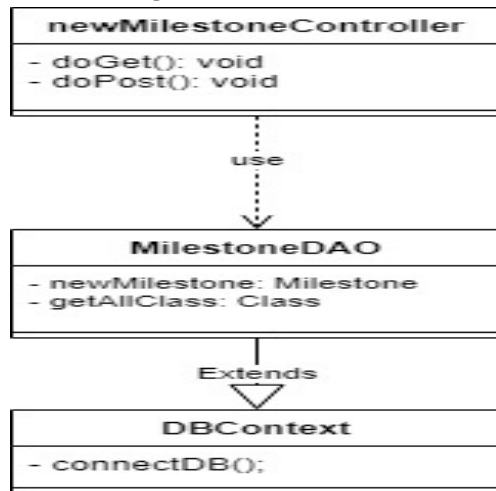
### d. Database Queries:

select \* from milestone



## 17. New Milestone:

### a. Class Diagram

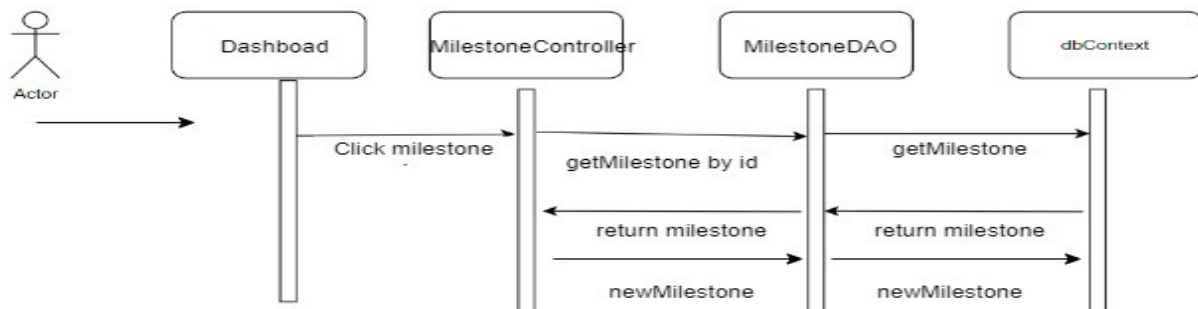


### b. Class Specifications

Class MilestoneDAO:

| No | Method       | Description          |
|----|--------------|----------------------|
| 1  | newMilestone | Create new Milestone |
| 2  | getAllClass  | Get all class        |

### c. Sequence Diagram(s)



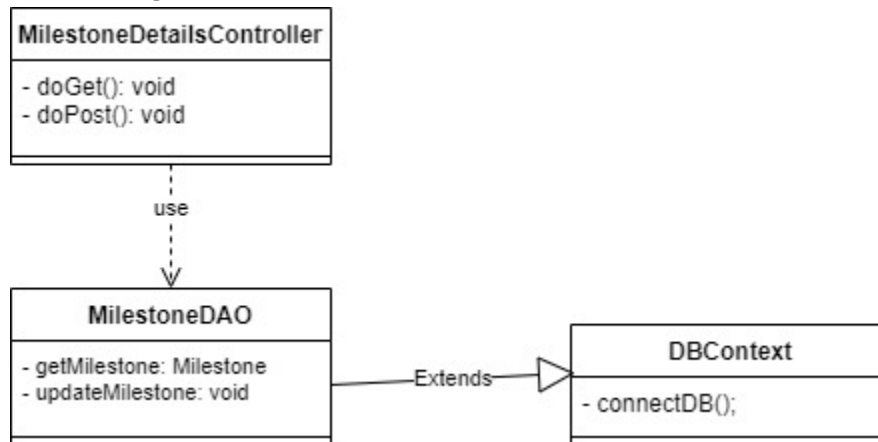
### d. Database Queries:

select \* from class

insert into milestone (?,?,?,?,?,?,?) values (?,?,?,?,?,?,?)

## 18. Milestone Details:

### a. Class Diagram

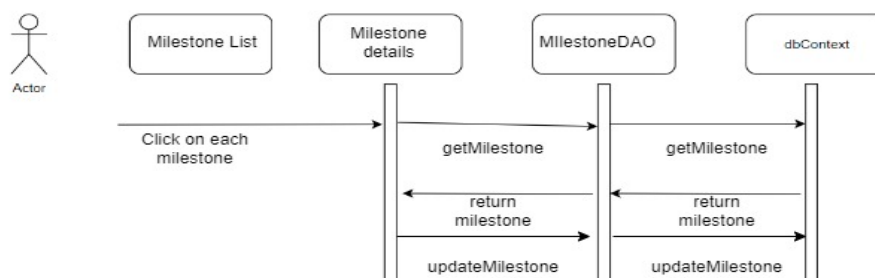


### b. Class Specifications

Class SubjectDAO:

| No | Method          | Description                   |
|----|-----------------|-------------------------------|
| 1  | getMilestone    | get milestone by milestone_id |
| 2  | updateMilestone | update information            |
| 3  | deleteMilestone | delete a milestone            |

### c. Sequence Diagram(s)



### d. Database Queries:

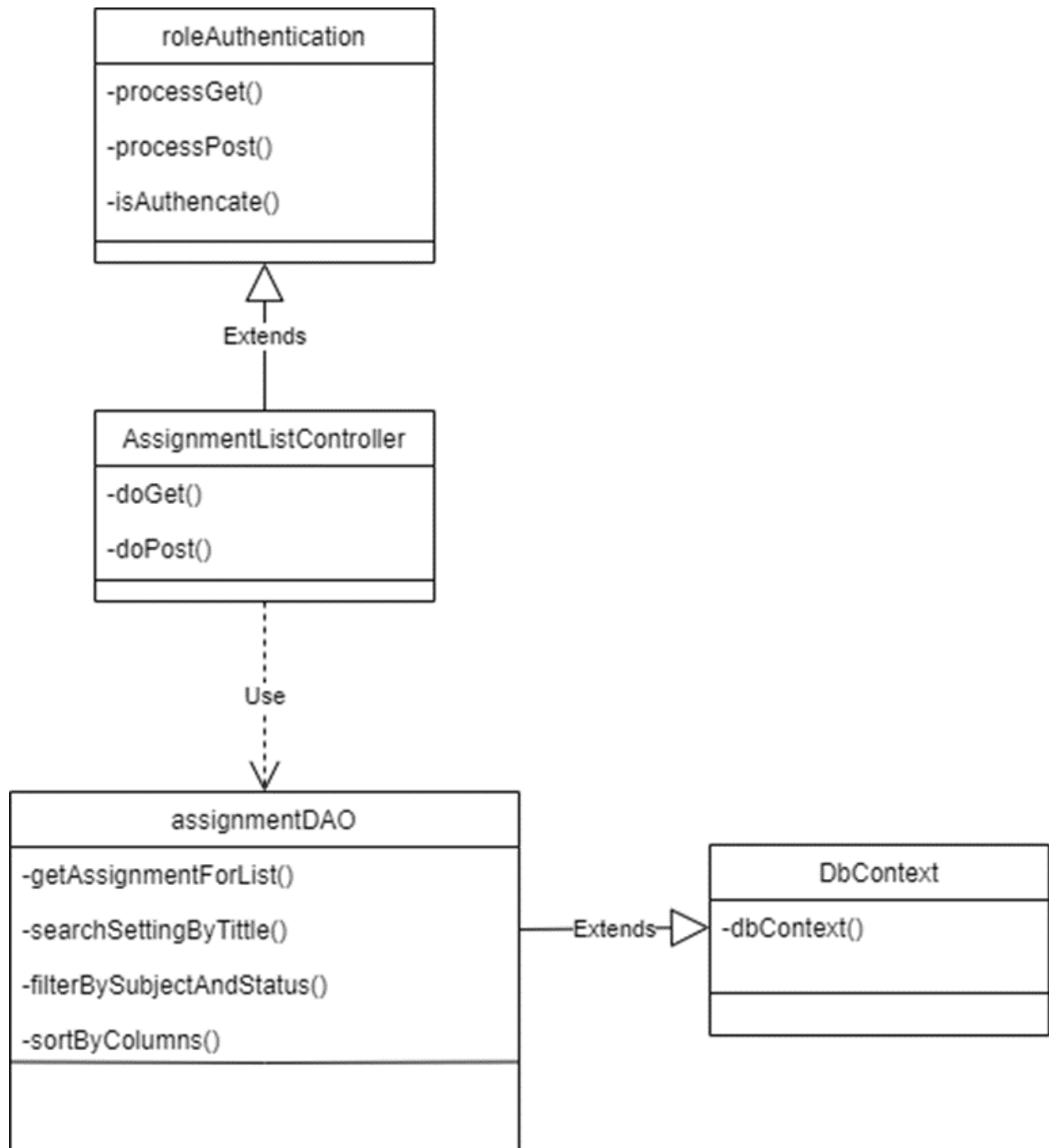
select \* from milestone where milestone\_id = ?

update milestone set ? = ? where milestone\_id = ?

delete from milestone where milestone\_id = ?

## 19. Assitment List

### a. Class Diagram



### b. Class Specifications

### roleAuthentication Class

| No | Method       | Description   |
|----|--------------|---|
| 01 | isAuthencate | Get the role of the account in session and check if the role matches or not |

### settingListControler Class

#### Class Methods

| No | Method          | Description  |
|----|-----------------|--|
| 01 | <i>doGet()</i>  | <i>Redirect</i>  |
| 02 | <i>doPost()</i> | <i>Send the all assignment List from database to frontend</i><br><i>Redirect</i> |

### settingDAOClass

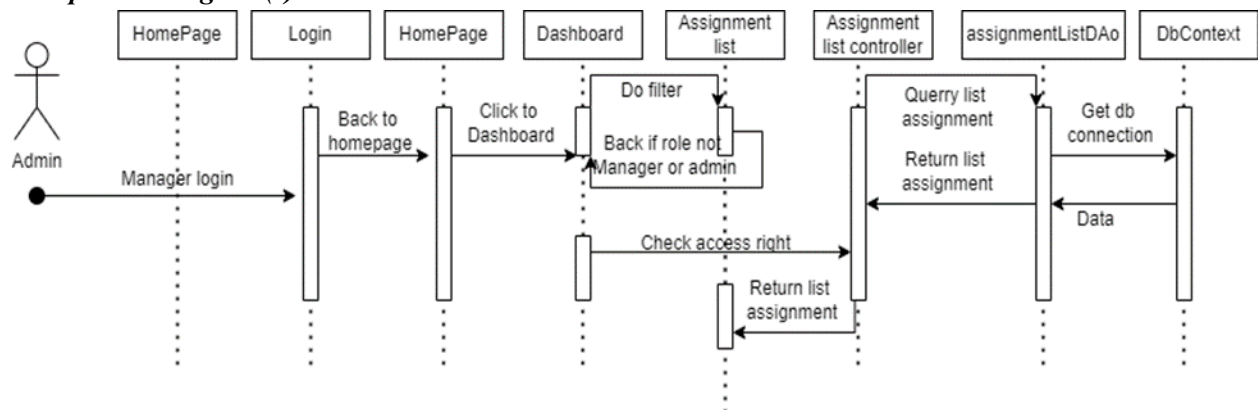
| No | Method                      | Description  |
|----|-----------------------------|--|
| 01 | getAssignmentForList()      | Get all assignment by paging from database                                       |
| 02 | searchSettingByTittle()     | Find and return assignment have tittle the same user input                       |
| 03 | filterBySubjectAndStatus () | Find and return assignment have subject and status the same user input in filter |

|    |                 |                      |
|----|-----------------|----------------------|
| 04 | sortByColumns() | Sort data by columns |
|    |                 |                      |

### *DbContext Class*

| No | Method       | Description                 |
|----|--------------|-----------------------------|
| 01 | -dbContext() | Connect program to database |

### *c. Sequence Diagram(s)*



### *d. Database Queries:*

`select * from assignment limit ? offset ?`

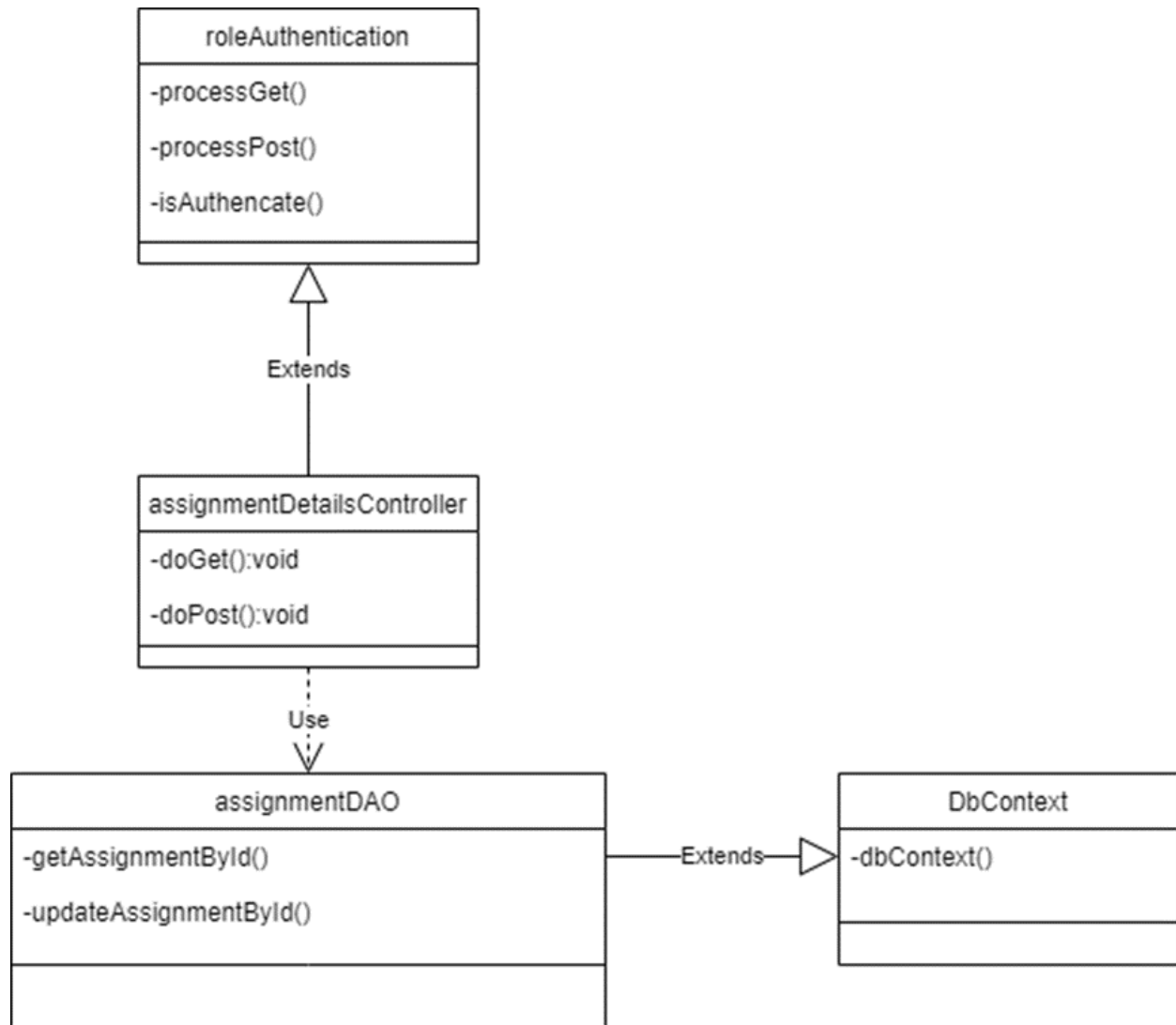
`select * from assignment order by " + type + " limit ? offset ?`

`select * from assignment where assignment_title like ? limit ? offset ?`

`select * from assignment where status = ? and subject = ?`

## 20. Assignment Details:

### *a. Class Diagram*



### *b. Class Specifications*

#### **roleAuthentication Class**

| No | Method       | Description   |
|----|--------------|---|
| 01 | isAuthencate | Get the role of the account in session and check if the role matches or not |

#### *settingDeatilsControler Class*

#### *Class Methods*

| No | Method   | Description   |
|----|----------|---|
| 01 | doGet()  | Redirect  |
| 02 | doPost() | Send assignment have the same Id from database to frontend<br>Redirec |

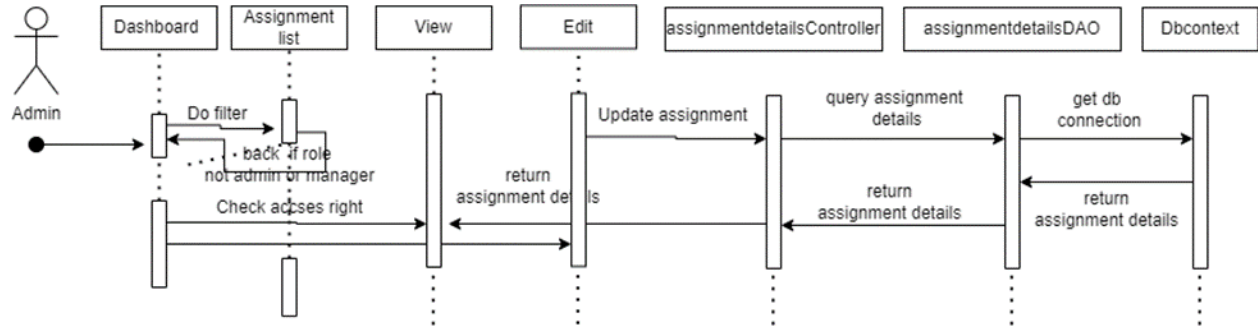
### *settingtDAOClass*

| No | Method                  | Description   |
|----|-------------------------|---|
| 01 | -getAssignmentById()    | Get assignment have the same Id from database setting from database |
| 02 | -updateAssignmentById() | Update assignment by id   |
|    |                         |   |

### *DbContext Class*

| No | Method       | Description                |
|----|--------------|----------------------------|
| 01 | -dbContext() | Connect program to databse |

### c. Sequence Diagram(s)

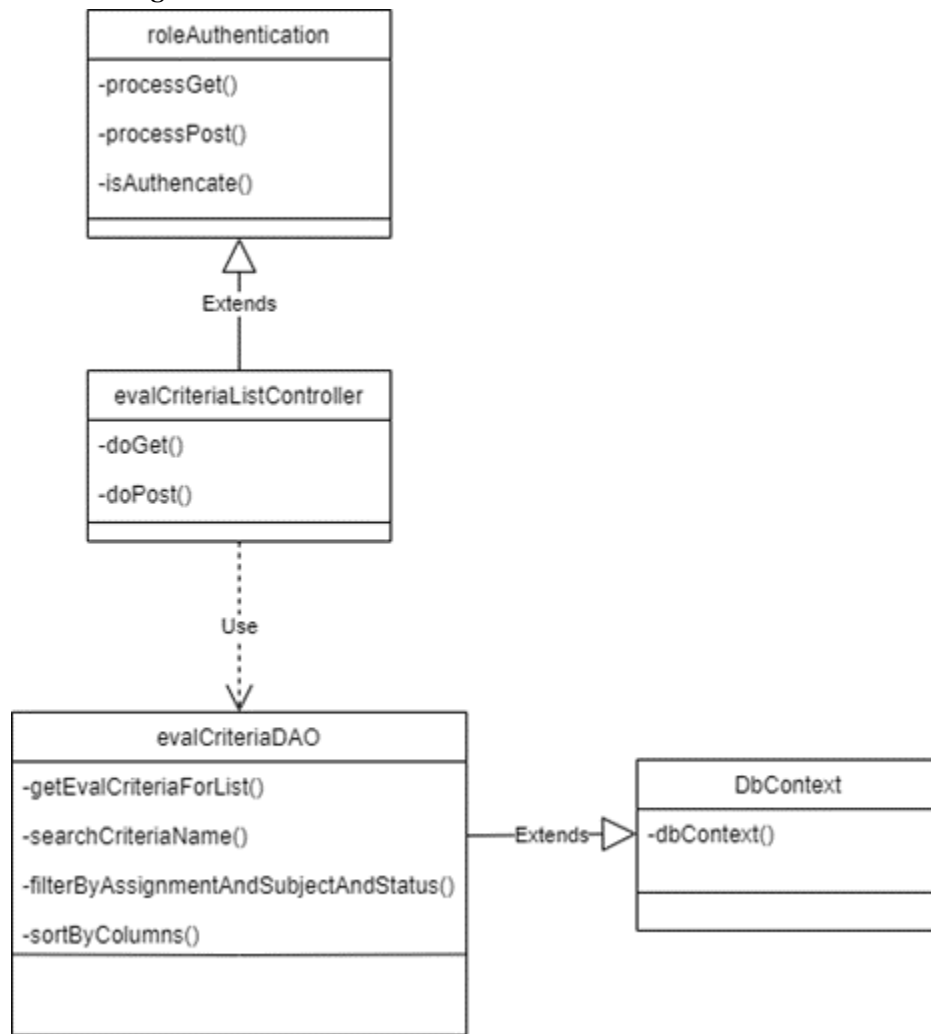


### d. Database Queries:

Select \* from assignment where assignment\_id = ?

## 21. Eval Criteria List:

### a. Class Diagram





### *b. Class Specifications*

#### **roleAuthentication Class**

| No | Method       | Description   |
|----|--------------|---|
| 01 | isAuthencate | Get the role of the account in session and check if the role matches or not |

#### *settingListControler Class*

##### *Class Methods*

| No | Method          | Description   |
|----|-----------------|---|
| 01 | <i>doGet()</i>  | <i>Redirect</i>   |
| 02 | <i>doPost()</i> | <i>Send the all eval criteria List from database to frontend</i><br><i>Redirect</i> |

#### *settingDAOClass*

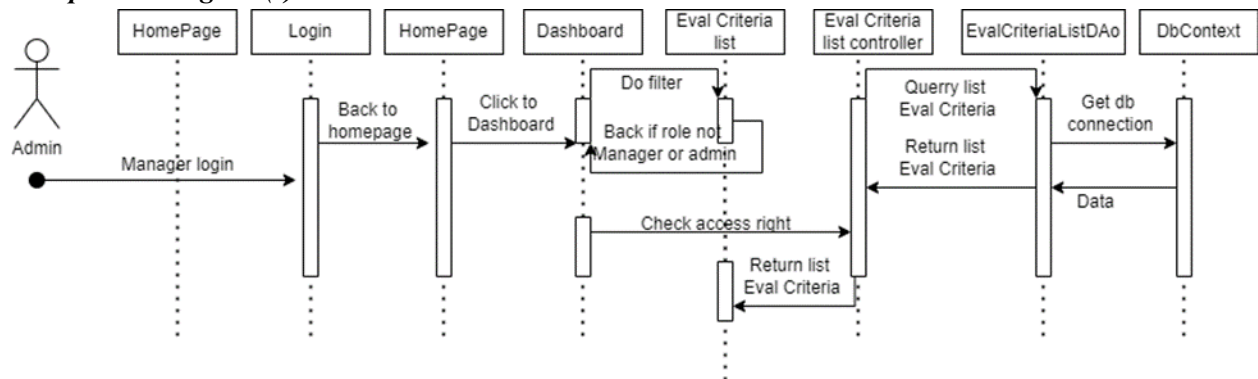
| No | Method                    | Description   |
|----|---------------------------|---|
| 01 | -getEvalCriteriaForList() | Get all eval criteria by paging from database               |
| 02 | -searchCriteriaName()     | Find and return eval criteria have name the same user input |

|    |  |   |
|----|--|---|
| 03 | -filterByAssignmentAndSubjectAndStatus() | Find and return eval criteria have assignment, subject and status the same user input in filter |
| 04 | sortByColumns()                          | Sort data by columns  |
|    |  |   |

### *DbContext Class*

| No | Method       | Description                 |
|----|--------------|-----------------------------|
| 01 | -dbContext() | Connect program to database |

### *c. Sequence Diagram(s)*



### *d. Database Queries:*

`select * from eval_criteria limit ? offset ?`

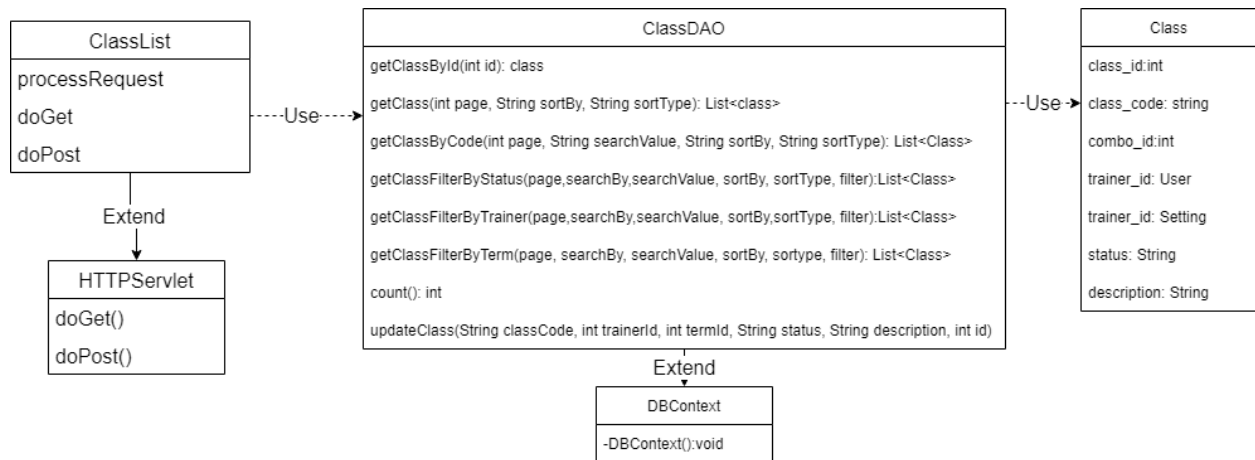
`select * from eval_criteria order by " + type + " limit ? offset ?`

`select * from eval_criteria where eval_criteria _name like ? limit ? offset ?`

`select * from eval_criteria where status = ? and subject = ? and assignment = ?`

## 22. Class List:

### a. Class Diagram



### b. Class Specifications

Servlet ClassList:

| No | Method   | Description                            |
|----|----------|--|
| 1  | doGet()  | Process and send data to classlist.jsp |
| 2  | doPost() |  |

Class ClassDAO:

| No | Method  | Description  |
|----|---|--|
| 1  | getClassById(int id): class                                       | Get class have class_Id= input id<br>And sort them by option of user |
| 2  | getClass(int page, String sortBy, String sortType): List<class>() | Get list class in page and sort them by option of user               |

|   |   |   |
|---|---|---|
| 3 | getClassByCode(int page, String searchValue, String sortBy, String sortType): List<Class>() | Get all class have code contain searchValue and sort them by option of user |
| 4 | getClassFilterByStatus(page,searchBy,searchValue, sortBy, sortType, filter):List<Class>()   | Filter classes by status:   |
| 5 | getClassFilterByTrainer(page, searchBy,searchValue, sortBy,sortType, filter):List<Class>    | Filter classes by trainer   |
| 6 | getClassFilterByTerm(page, searchBy, searchValue, sortBy, sortype, filter): List<Class>     | Filter classes by term  |
| 7 | count(): int  | Count and return the total number of page.                                  |
| 8 | updateClass( classCode, trainerId, termId, String, String description, int id)              | Update a class.   |

Class DBContext: Connect to database

Class Class: Contains the information of class

### ***c. Sequence Diagram(s)***

### ***d. Database Queries:***

## 23. Class details:

### a. Class Diagram

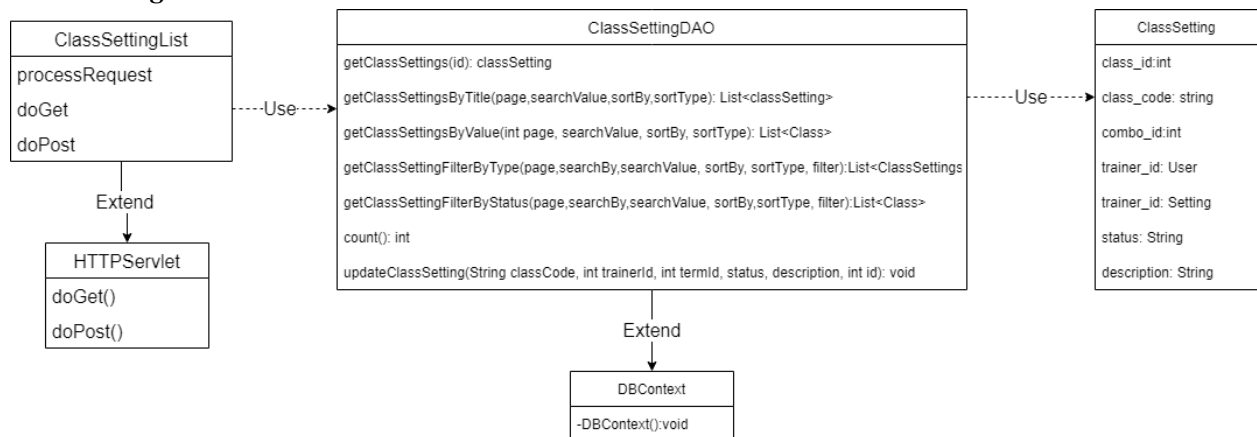
### b. Class Specifications

### c. Sequence Diagram(s)

### d. Database Queries:

## 24. Class Setting List:

### a. Class Diagram



### b. Class Specifications

Servlet **ClassSettingList**:

| No | Method                | Description  |
|----|-----------------------|--|
| 1  | <code>doGet()</code>  | Process and send data to <code>classsettinglist.jsp</code> |
| 2  | <code>doPost()</code> |  |

Class **ClassSettingListDAO**:

| No | Method                            | Description  |
|----|-----------------------------------|--|
| 1  | <code>getClassSettings(id)</code> | Get class setting have <code>id= searchValue</code><br>And sort them by option of user |

|   |   |   |
|---|---|---|
| 2 | getClassSettingsByTitle(page, searchValue, sortBy, sortType):   | Get contacts have title contains searchValue<br>And sort them by option of user   |
| 3 | getClassSettingsByValue(int page, searchValue, sortBy, sortType): List<Class>                           | Get contacts have value contains the search value and sort them by option of user |
| 4 | getClassSettingFilterByType(page, searchBy, searchValue, sortBy, sortType, filter): List<ClassSettings> | filter classsetting by type, and sort them by option of user                      |
| 5 | getClassSettingFilterByStatus(page, searchBy, searchValue, sortBy, sortType, filter)                    | filter classsetting by status, and sort them by option of user                    |
| 6 | count()   | Count the number of class setting and return maximum of page number               |
| 7 | updateClassSetting(String classCode, int trainerId, int termId, status, description, int id)            | Update a class setting.   |

Class DBContext: Connect to database

Class Classsetting: Contains the information class setting

**c. Sequence Diagram(s)**

**d. Database Queries:**

SELECT \* FROM class\_setting WHERE setting\_value like ? and status=? ORDER BY type\_id\_fk desc LIMIT 5 OFFSET ?

SELECT \* FROM class\_setting WHERE setting\_value like ? and status=? ORDER BY type\_id\_fk asc LIMIT 5 OFFSET ?

SELECT \* FROM class\_setting WHERE setting\_value like ? and status=? ORDER BY setting\_title desc LIMIT 5 OFFSET ?

SELECT \* FROM class\_setting WHERE setting\_value like ? and status=? ORDER BY setting\_title asc LIMIT 5 OFFSET ?

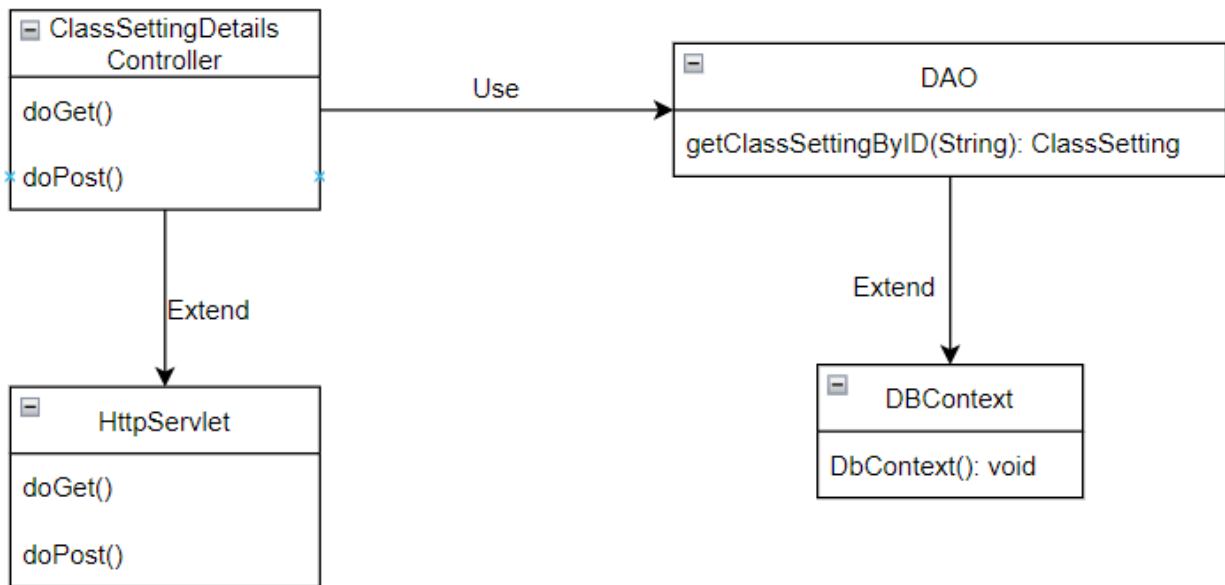
SELECT \* FROM class\_setting WHERE setting\_value like ? and status=? ORDER BY setting\_value asc LIMIT 5 OFFSET ?

SELECT \* FROM class\_setting WHERE setting\_value like ? and status=? ORDER BY setting\_value desc LIMIT 5 OFFSET ?

SELECT \* FROM class\_setting WHERE setting\_value like ? and status=? ORDER BY display\_order asc LIMIT 5 OFFSET ?

## 25. Class Setting Details:

### a. Class Diagram:

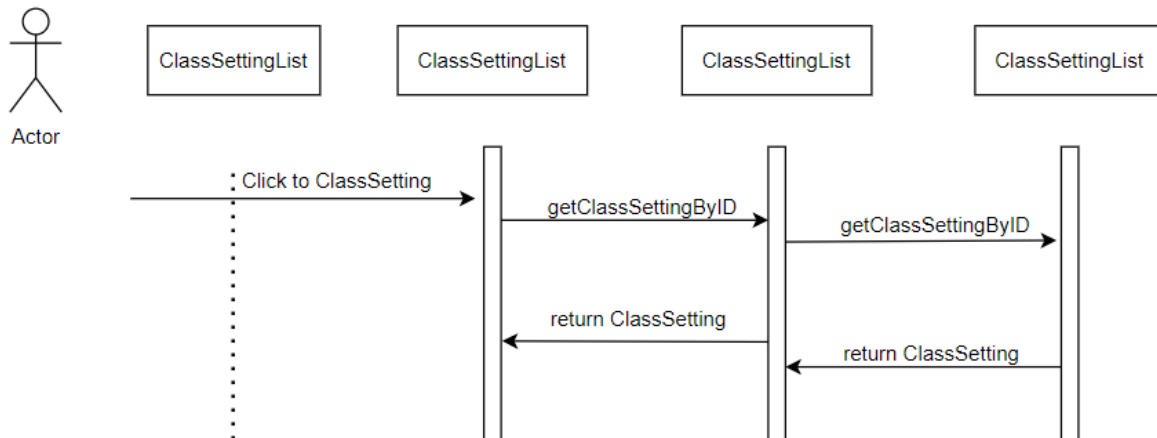


### b. Class Specifications:

Class ClassSettingDAO:

| No | Method              | Description                                  |
|----|---------------------|--|
| 1  | getClassSettingByID | Get a subject have seting_ID equals to input |

### c. Sequence Diagram(s):



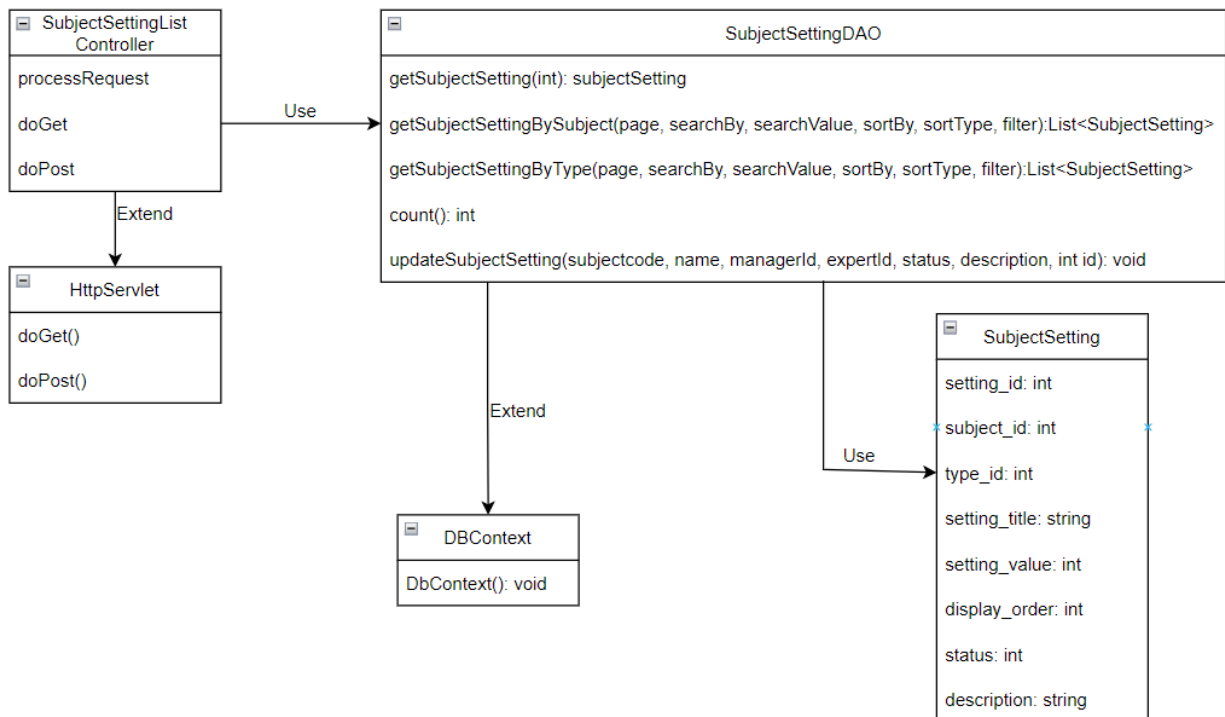
### d. Database Queries:

Select \* from class\_setting where setting\_id = ? order by setting\_id desc LIMIT 5 OFFSET ?

Select \* from class\_setting where setting\_id = ? order by setting\_id asc LIMIT 5 OFFSET ?

## 26. Subject Setting List:

### a. Class Diagram:





***b. Class Specifications:***

SubjectSettingList:

| No | Method   | Description                                   |
|----|----------|---|
| 1  | doGet()  | Process and send data to classsettinglist.jsp |
| 2  | doPost() |   |

SubjectSettingListDAO:

| No | Method   | Description  |
|----|--|--|
| 1  | getSubjectSetting(id)  | Get subject setting have id = searchValue<br>And sort them by option of user |
| 2  | getSubjectSettingBySubject(page,searchBy,searchValue,sortBy, sortType, filter):List<SubjectSettings> | filter subject setting by subject, and sort them by option of user           |
| 3  | getSubjectSettingByType(page,searchBy,searchValue,sortBy,sortType, filter)                           | filter subject setting by type, and sort them by option of user              |
| 6  | count()  | Count the number of subject setting and return maximum of page number        |
| 7  | updateSubjectSetting(subjectcode, name, managerId, expertId, status, description, int id)            | Update a subject setting.  |

DBContext: Connect to database

SubjectSetting: Contains the information subject setting

***c. Sequence Diagram(s):***

***d. Database Queries:***

```
SELECT * FROM subject_setting WHERE setting_value like ? and status=? ORDER BY type_id desc LIMIT 5  
OFFSET ?
```

```
SELECT * FROM subject_setting WHERE setting_value like ? and status=? ORDER BY type_id asc LIMIT 5  
OFFSET ?
```

```
SELECT * FROM subject_setting WHERE setting_value like ? and status=? ORDER BY subject_id desc  
LIMIT 5 OFFSET ?
```

```
SELECT * FROM subject_setting WHERE setting_value like ? and status=? ORDER BY subject_id asc LIMIT  
5 OFFSET ?
```

**27. Subject Setting Details:**

***a. Class Diagram:***

***b. Class Specifications:***

***c. Sequence Diagram(s):***

***d. Database Queries:***

**28. System Permission:**

***a. Class Diagram:***

***b. Class Specifications:***

***c. Sequence Diagram(s):***

***d. Database Queries:***

**29. User Authorization:**

***a. Class Diagram:***

***b. Class Specifications:***

*c. Sequence Diagram(s):*

*d. Database Queries:*

**30. Eval-Criteria Details:**

*a. Class Diagram:*

*b. Class Specifications:*

*c. Sequence Diagram(s):*

*d. Database Queries:*

**31. Class Eval-Criteria List:**

*a. Class Diagram:*

*b. Class Specifications:*

*c. Sequence Diagram(s):*

*d. Database Queries:*

**32. Class Eval-Criteria Details:**

*a. Class Diagram:*

*b. Class Specifications:*

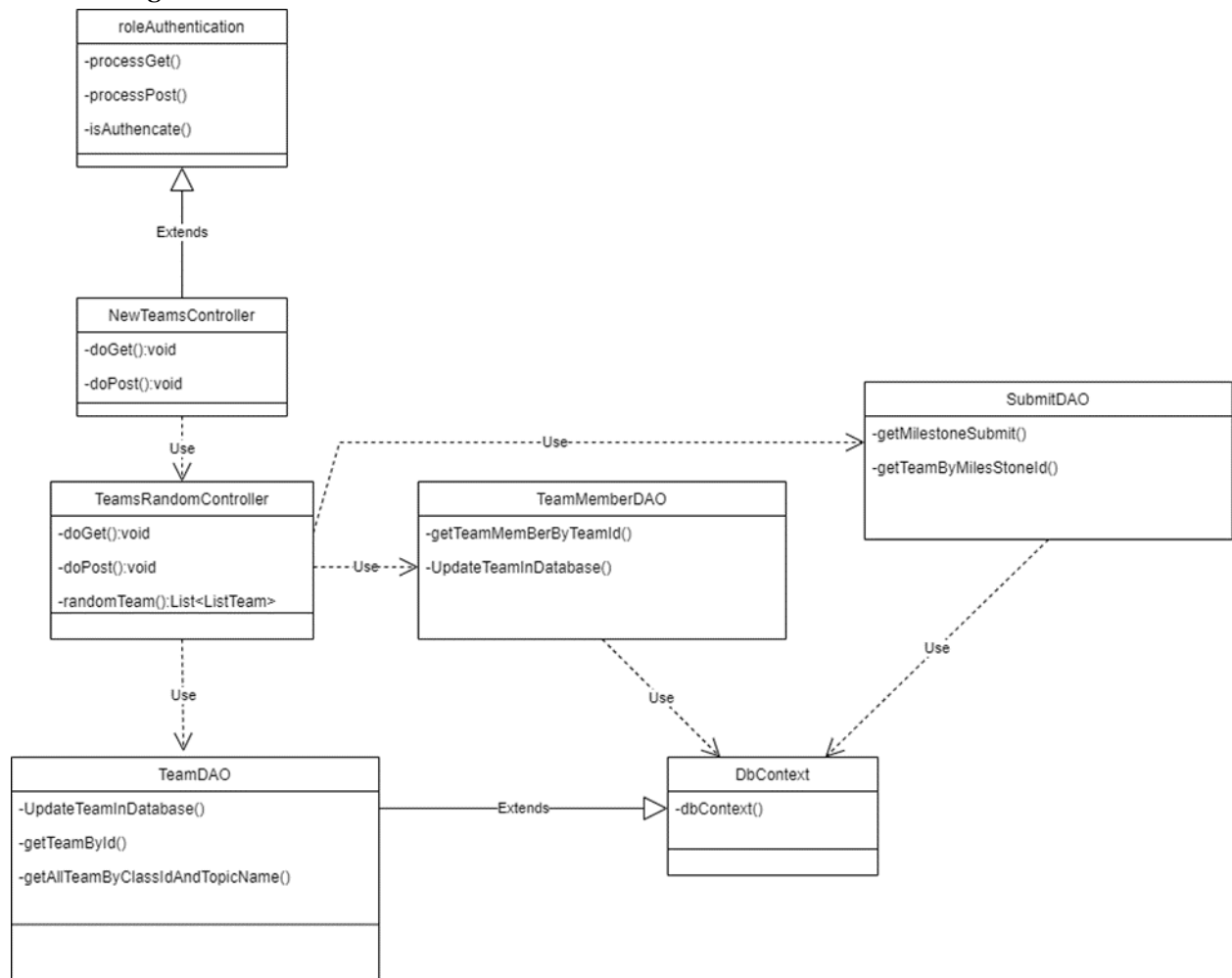
*c. Sequence Diagram(s):*

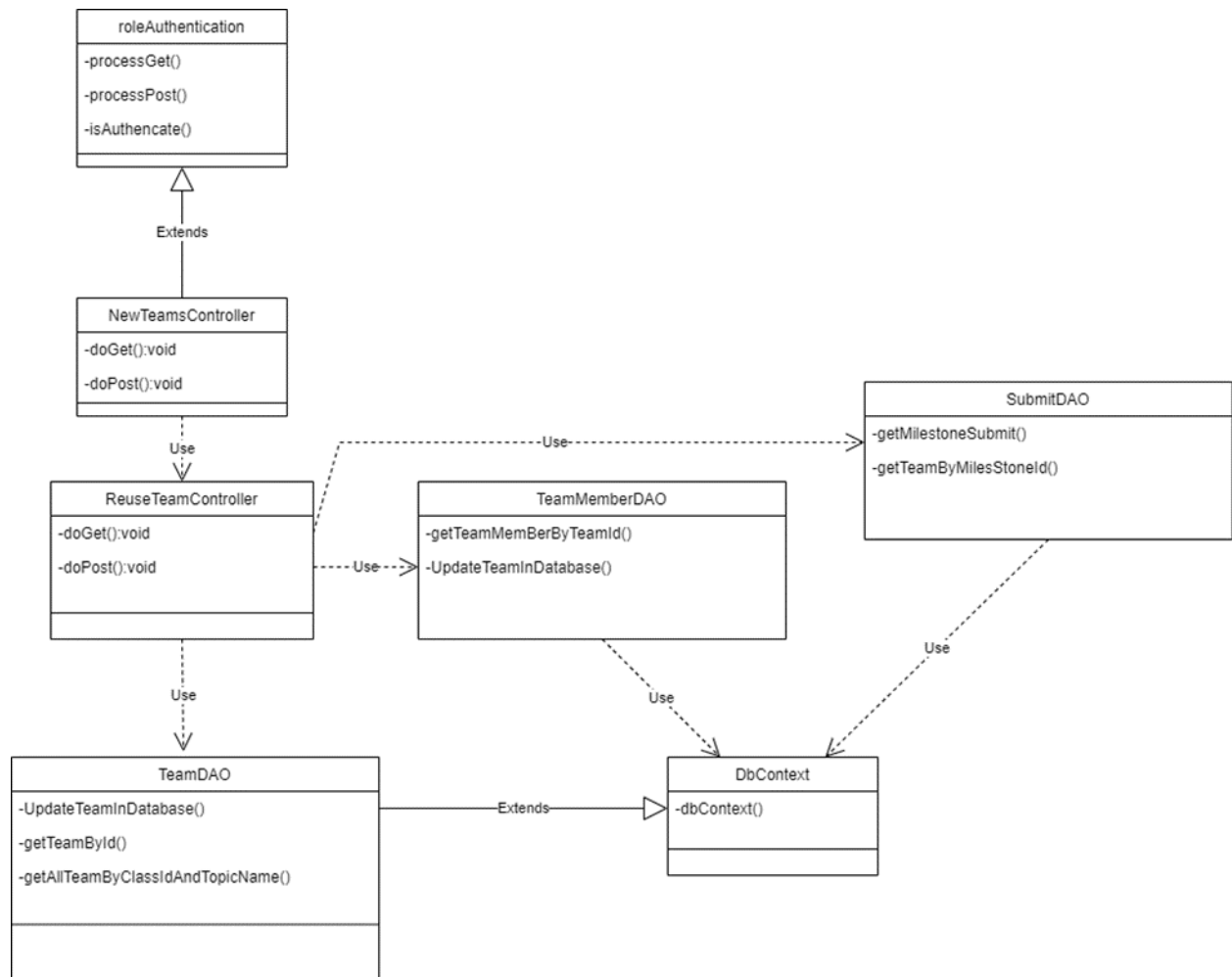
*d. Database Queries:*

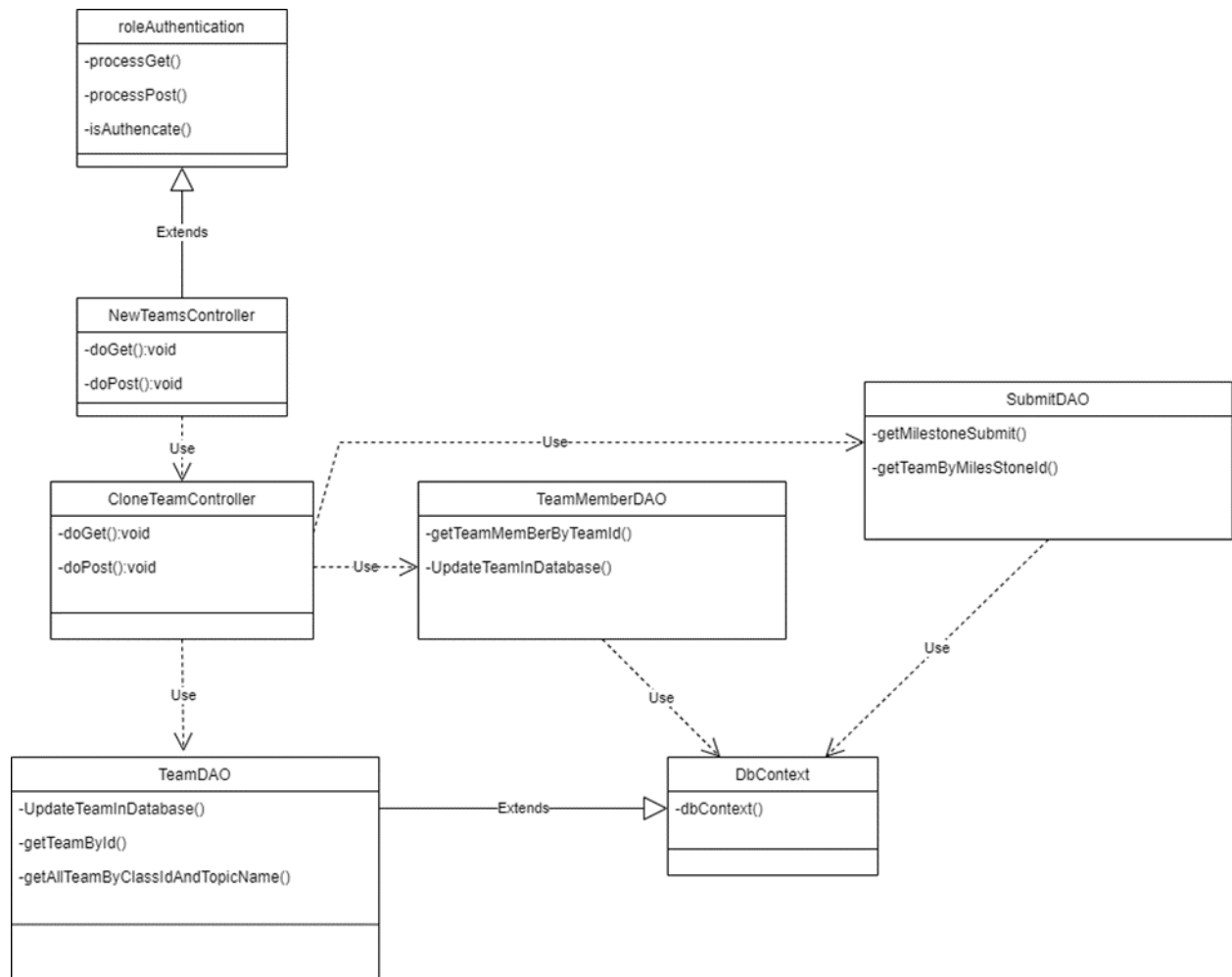
33.New Teams

### 33.New Teams:

#### a. Class Diagram:







## b. Class Specifications:

### *NewTeamsControllerClass*

#### *Class Methods*

| No | Method          | Description                               |
|----|-----------------|---|
| 01 | <i>doGet()</i>  | <i>Send the number student in a class</i> |
| 02 | <i>doPost()</i> |   |

### *TeamsRandomControllerClass*

| No | Method | Description |
|----|--------|-------------|
|----|--------|-------------|

|    |              |  |
|----|--------------|--|
| 01 | doGet()      | Save change the team create by rom to database. Send the data to team list   |
| 02 | doPost()     | Send the team create by random to the newteams.jsp for display the information of the team   |
| 03 | randomTeam() | Random the user of class to the team list. And the data pf the randomteam is a user of class in a random team order from top to bottom |
|    |              |  |

#### ***TeamMemberDAO***

| No | Method                   | Description                      |
|----|--------------------------|----------------------------------|
| 01 | -getTeamMemBerByTeamId() | Get the user of the team         |
| 02 | -UpdateTeamInDatabase()  | Save the member team to database |

#### ***ReuseTeamDAO***

| No | Method     | Description  |
|----|------------|--|
| 01 | -doGet ()  | Sen to the team list   |
| 02 | -doPost () | Get team and team member already already in the database of a class in an assignment |

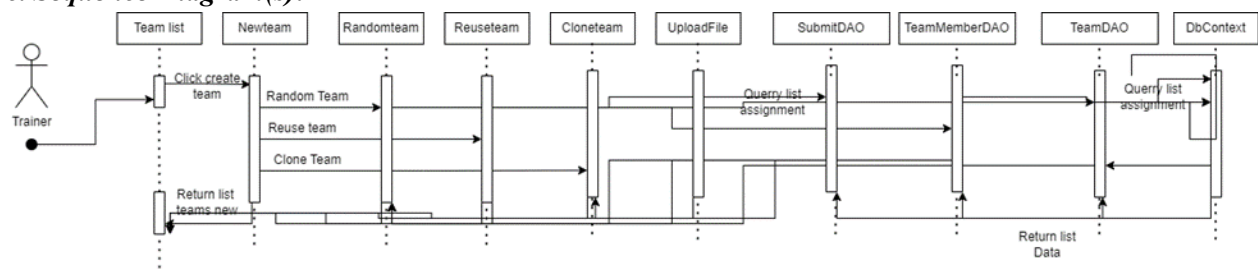
### CloneTeamDAO

| No | Method     | Description  |
|----|------------|--|
| 01 | -doGet ()  | Save the team and member team to the database, create a new record in data base like a old team but change the team name and update topic name |
| 02 | -doPost () | Get team and team member already already in the database of a class in an assignment   |

### TeamDAO

| No | Method                             | Description                                 |
|----|------------------------------------|---|
| 01 | -UpdateTeamInDatabase()            | Save the team to database                   |
| 02 | -getTeamById()                     | Get the team id for User of team            |
| 03 | -getAllTeamByClassIdAndTopicName() | Get the team by class id ang the topic name |

### c. Sequence Diagram(s):



### d. Database Queries:

`select * from class_user`



```
select * from submit where milestone_id_su_fk =?
```

```
select distinct milestone_id_su_fk from submit
```

```
INSERT INTO `swp391`.`team_member`\n
```

```
  + "("team_id_tm`,\n
```

```
  + "`user_id_tm`,\n
```

```
  + "`is_leader`,\n
```

```
  + "`is_active`)\n
```

```
  + "VALUES\n
```

```
  + "(?,?,?,?);"
```

```
SELECT * FROM swp391.team_member where team_id_tm = ?
```

```
INSERT INTO `swp391`.`team`\n
```

```
  + "("class_id_team`,\n
```

```
  + "`team_code`,\n
```

```
  + "`topic_name`,\n
```

```
  + "`status`,\n
```

```
  + "`description`)\n
```

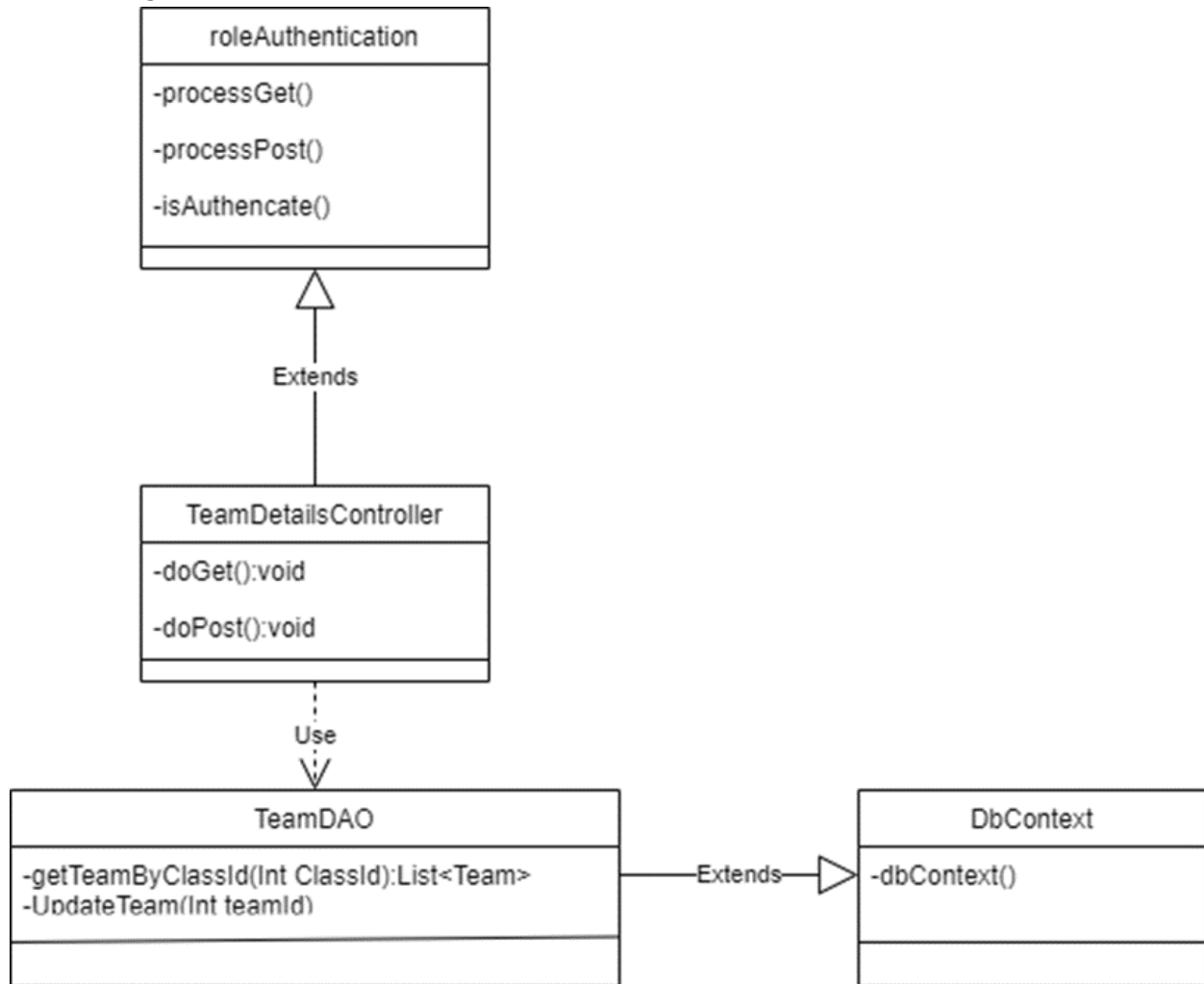
```
  + "VALUES\n
```

```
  + "(?,?,?,?);"
```

```
SELECT * FROM swp391.team where team_id = ?
```

### 34.Team details:

#### a. Class Diagram:



#### b. Class Specifications:

##### roleAuthentication Class

| No | Method       | Description   |
|----|--------------|---|
| 01 | isAuthencate | Get the role of the account in session and check if the role matches or not |

##### TeamDeatilsControler Class

##### Class Methods

| No | Method   | Description   |
|----|----------|---|
| 01 | doGet()  | Redirect  |
| 02 | doPost() | Send the all settingList have the same settingId from database to frontend<br><br>Redirec |

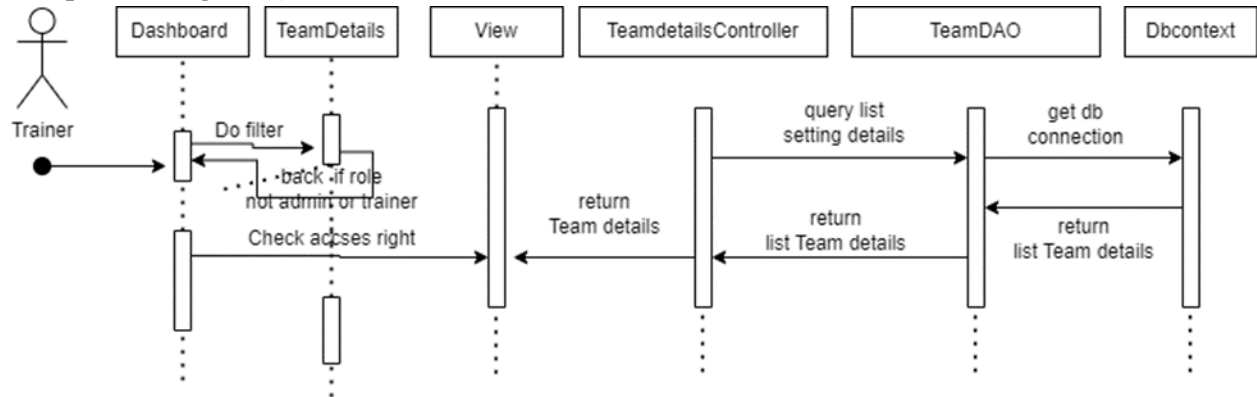
#### *TeamDAOClass*

| No | Method                  | Description  |
|----|-------------------------|--|
| 01 | getSettingByTypeId()    | Get all TeamList have the same ClassId from database setting from database |
| 02 | -UpdateTeam(Int teamId) | Update team id   |
|    |                         |  |

#### *DbContext Class*

| No | Method       | Description                |
|----|--------------|----------------------------|
| 01 | -dbContext() | Connect program to databse |

**c. Sequence Diagram(s):**



**d. Database Queries:**

Select \* from team where class\_id = ?

UPDATE `swp391`.`team`

SET

`team\_id` = ?,

`class\_id\_team` = ?,

`team\_code` = ?,

`topic\_name` = ?,

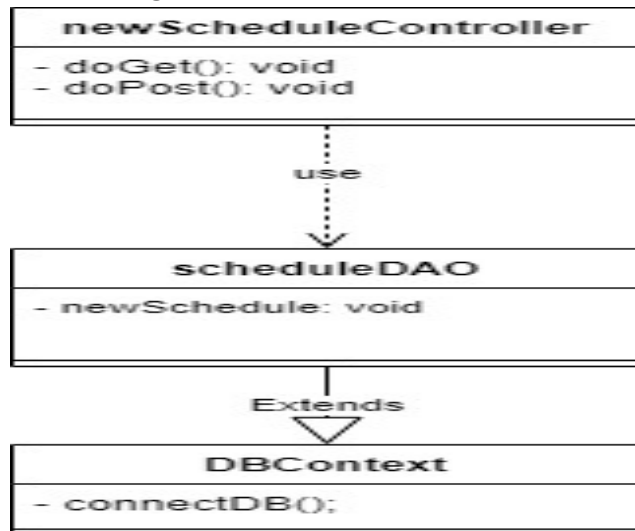
`status` = ?,

`description` = ?,

WHERE `team\_id` = ?

### 35. Schedule details:

#### a. Class Diagram

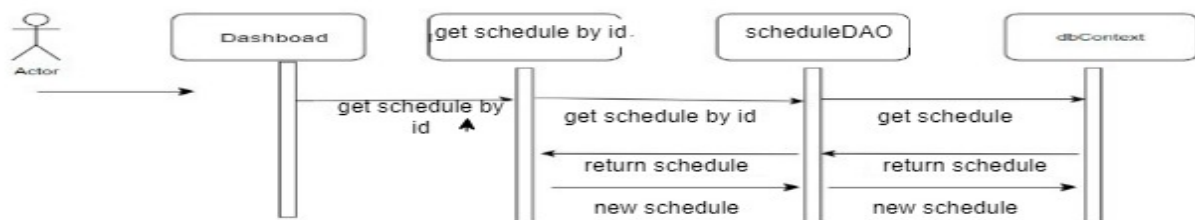


#### b. Class Specifications

Class scheduleDAO:

| No | Method      | Description         |
|----|-------------|---------------------|
| 1  | newSchedule | Create new Schedule |

#### c. Sequence Diagram(s)

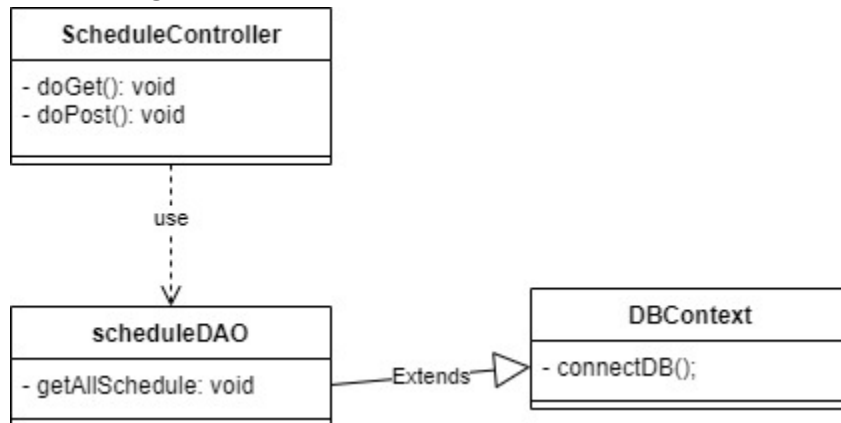


#### d. Database Queries:

insert into schedule(?,?,?,?,?,?,?) values (?,?,?,?,?,?,?)

### 36. Schedule List:

#### a. Class Diagram

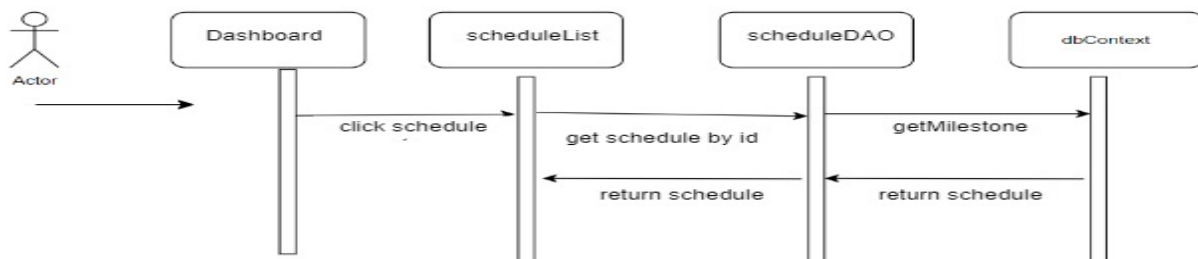


#### b. Class Specifications

Class **scheduleDAO**:

| No | Method         | Description                   |
|----|----------------|-------------------------------|
| 1  | getAllSchedule | list all schedule with paging |

#### c. Sequence Diagram(s)

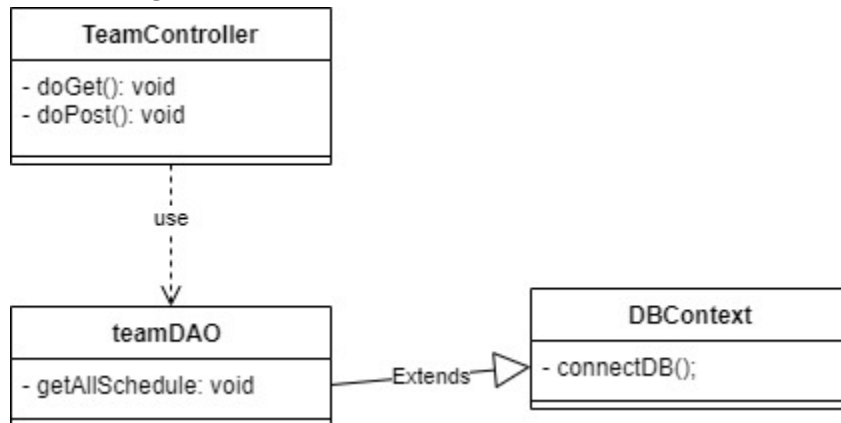


#### d. Database Queries:

select \* from schedule

### 37. Team List:

#### a. Class Diagram

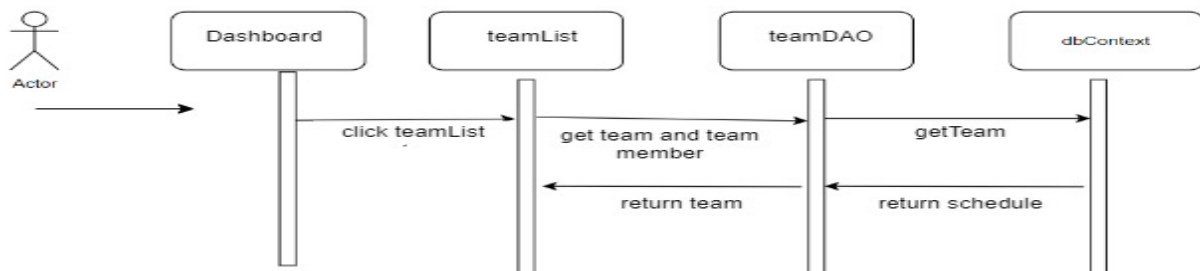


#### b. Class Specifications

Class TeamDAO:

| No | Method     | Description  |
|----|------------|--------------|
| 1  | getAllteam | Get all team |

#### c. Sequence Diagram(s)

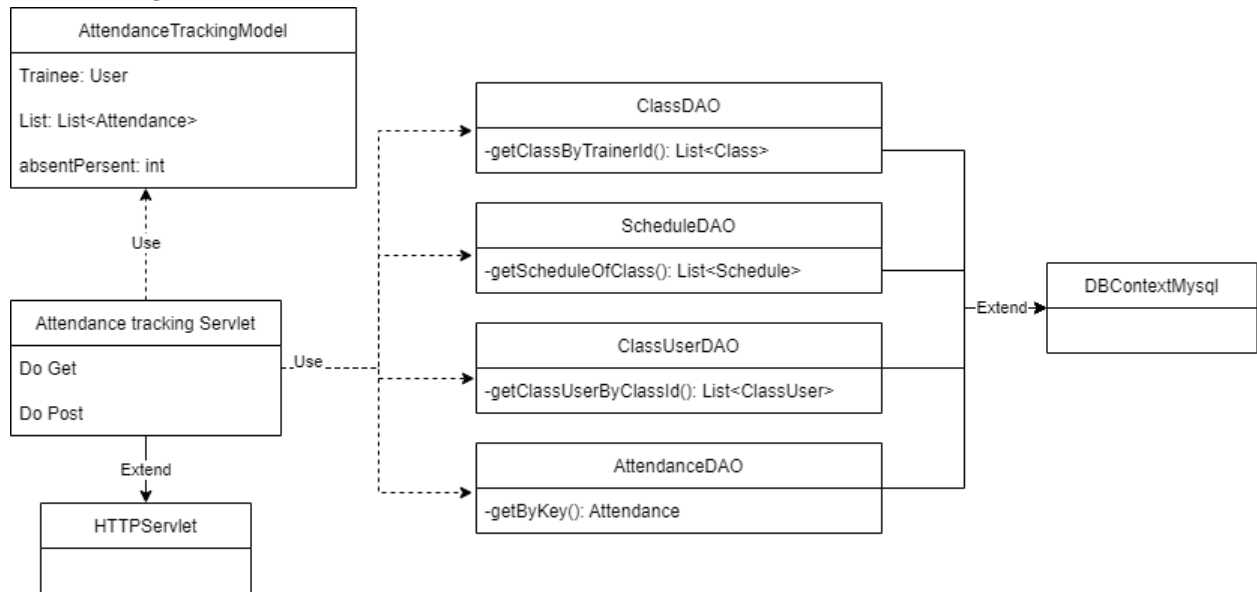


#### d. Database Queries:

select \* from team

### 38. Attendance Tracking:

#### a. Class Diagram



#### b. Class Specifications

Class AttendanceDAO:

| No | Method   | Description                                   |
|----|----------|---|
| 1  | getByKey | get attendance by, class schedule and trainee |

Class ScheduleDAO:

| No | Method             | Description                           |
|----|--------------------|---------------------------------------|
| 1  | getScheduleOfClass | get all schedule of a class by its id |

Class ClassUserDAO:

| No | Method                | Description                   |
|----|-----------------------|-------------------------------|
| 1  | getClassUserByClassId | Get all trainee id of a class |



Class ClassDAO:

| No | Method              | Description                |
|----|---------------------|----------------------------|
| 1  | getClassByTrainerId | Get all class of a trainer |

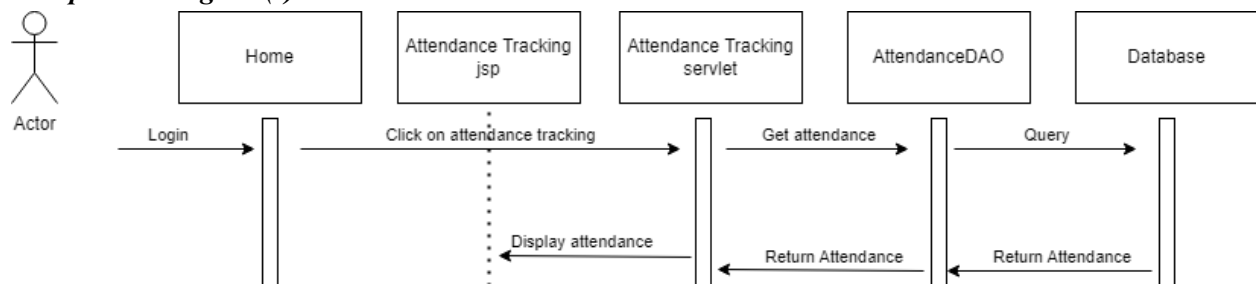
Attendance tracking servlet:

| No | Method | Description                             |
|----|--------|---|
| 1  | doGet  | Get, process data and send to front end |
| 2  | doPost |   |

AttendanceTrackingModel: Contain information of trainee, its attendance in all schedule of a class and the absent percentage

| No | Properties     | Description  |
|----|----------------|--|
| 1  | user           | All information of user                            |
| 2  | list           | Attendance of this user in all schedule of a class |
| 3  | absent percent | Absent percent of user                             |

### c. Sequence Diagram(s)



### d. Database Queries:

select \* from attendance where class\_id\_att=? and schedule\_id\_att=?

select \* from attendance where class\_id\_att=? and schedule\_id\_att=? and trainee\_id\_att=?

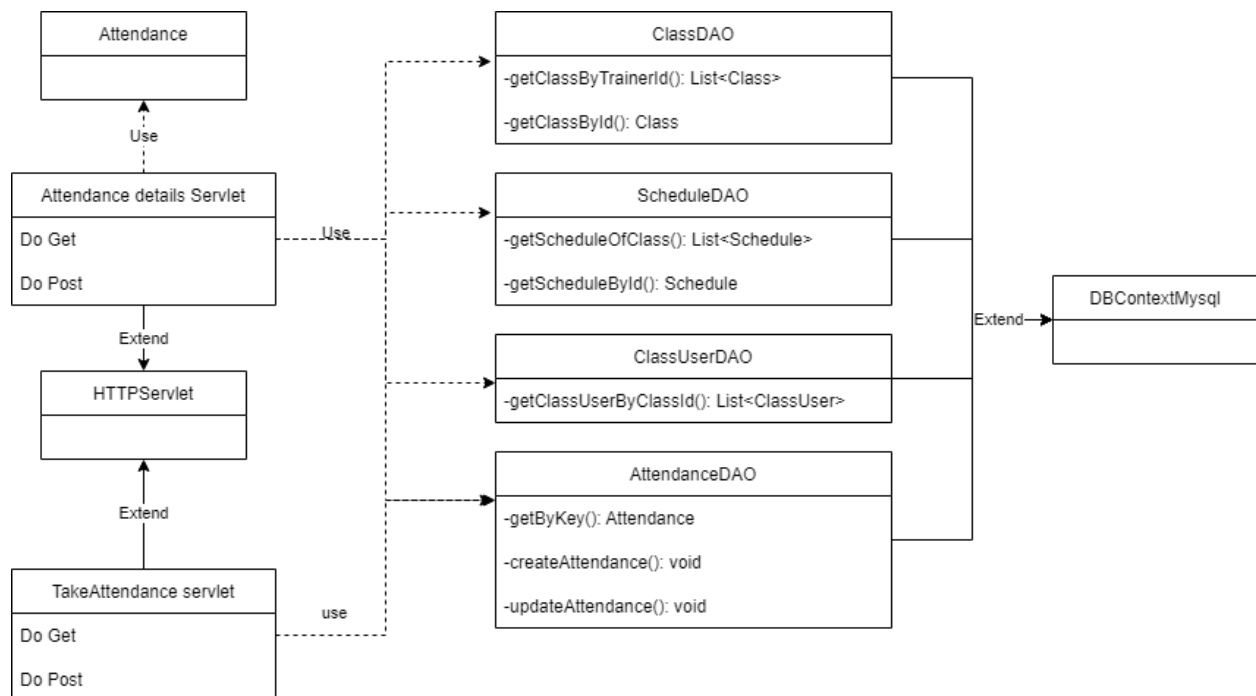
select \* from class where trainer\_id=?

select \* from class\_user where class\_user\_id\_fk=?

select \* from schedule where class\_id\_schedule=?

### 39. Attendance Details:

#### a. Class Diagram



#### b. Class Specifications

Class AttendanceDAO:

| No | Method           | Description                                   |
|----|------------------|---|
| 1  | getByKey         | get attendance by, class schedule and trainee |
| 2  | createAttendance | create new row in table attendance            |
| 3  | updateAttendance | Update an attendance                          |

Class ScheduleDAO:

| No | Method             | Description                           |
|----|--------------------|---------------------------------------|
| 1  | getScheduleOfClass | Get all schedule of a class by its id |
| 2  | getScheduleById    | Get schedule by its id                |

Class ClassUserDAO:

| No | Method                | Description                   |
|----|-----------------------|-------------------------------|
| 1  | getClassUserByClassId | Get all trainee id of a class |

Class ClassDAO:

| No | Method              | Description                |
|----|---------------------|----------------------------|
| 1  | getClassByTrainerId | Get all class of a trainer |
| 2  | getClassById        | Get class by its id        |

Attendance Details servlet:

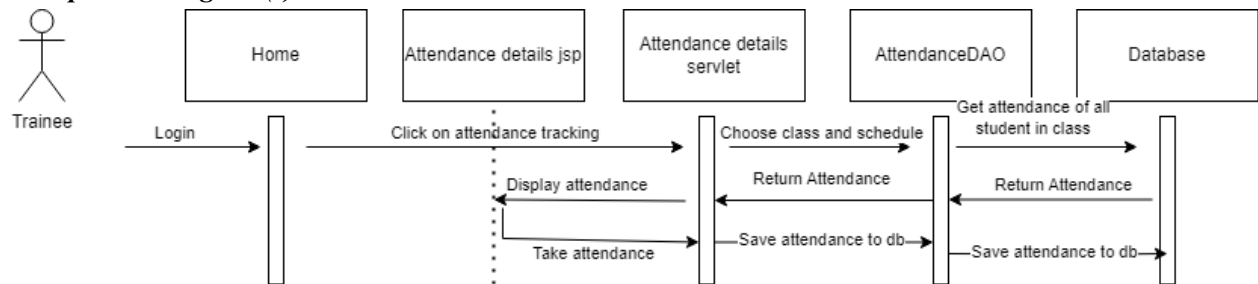
| No | Method | Description                             |
|----|--------|---|
| 1  | doGet  | Get, process data and send to front end |
| 2  | doPost |   |

Take Attendance servlet:

| No | Method | Description |
|----|--------|-------------|
| 1  | doGet  |             |

|   |        |   |
|---|--------|---|
| 2 | doPost | Get data from form and update to database |
|---|--------|---|

### c. Sequence Diagram(s)



### d. Database Queries:

select \* from attendance where class\_id\_att=? and schedule\_id\_att=?

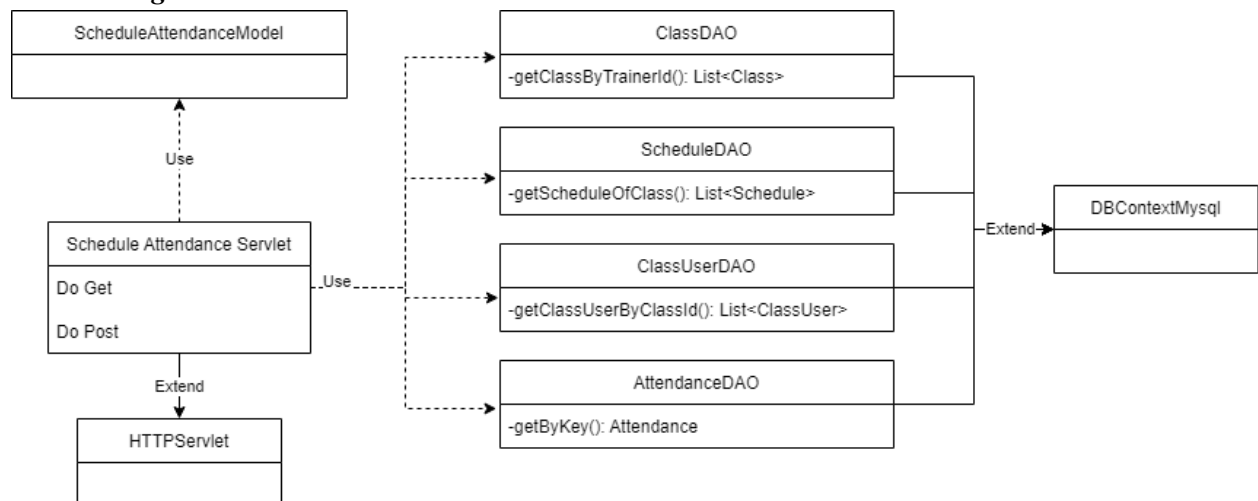
select \* from attendance where class\_id\_att=? and schedule\_id\_att=? and trainee\_id\_att=?

update attendance set status=?, comment=? where class\_id\_att=? and schedule\_id\_att=? and trainee\_id\_att=?

insert into attendance (class\_id\_att,schedule\_id\_att,trainee\_id\_att,status,comment) value (?,?,?,?/?)

## 40. Schedule Attendance:

### a. Class Diagram



### ***b. Class Specifications***

Class AttendanceDAO:

| No | Method   | Description                                   |
|----|----------|---|
| 1  | getByKey | get attendance by, class schedule and trainee |

Class ScheduleDAO:

| No | Method             | Description                           |
|----|--------------------|---------------------------------------|
| 1  | getScheduleOfClass | get all schedule of a class by its id |

Class ClassUserDAO:

| No | Method                | Description                   |
|----|-----------------------|-------------------------------|
| 1  | getClassUserByClassId | Get all trainee id of a class |

Class ClassDAO:

| No | Method              | Description                |
|----|---------------------|----------------------------|
| 1  | getClassByTrainerId | Get all class of a trainer |

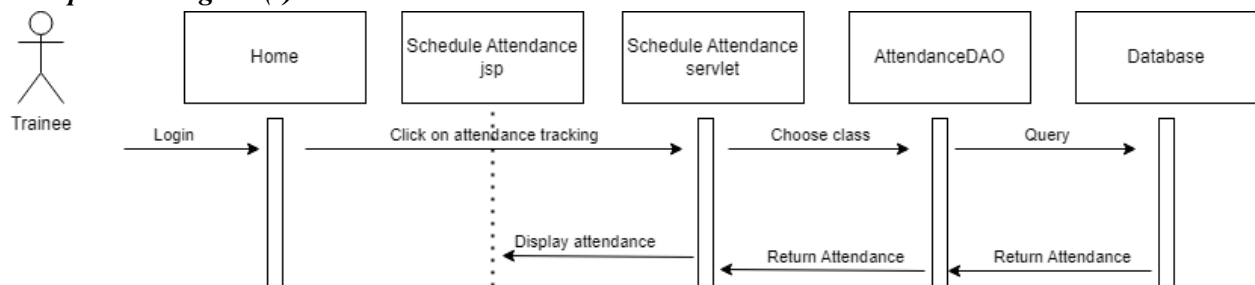
Schedule Attendance servlet:

| No | Method | Description                             |
|----|--------|---|
| 1  | doGet  | Get, process data and send to front end |
| 2  | doPost |   |

ScheduleAttendanceModel: Contain information of attendance and schedule

| No | Properties    | Description |
|----|---------------|-------------|
| 1  | slot          |             |
| 2  | training_date |             |
| 3  | from_time     |             |
| 4  | to_time       |             |
| 5  | room          |             |
| 6  | attendance    |             |
| 7  | comment       |             |

**c. Sequence Diagram(s)**



**d. Database Queries:**

select \* from attendance where class\_id\_att=? and schedule\_id\_att=?

select \* from attendance where class\_id\_att=? and schedule\_id\_att=? and trainee\_id\_att=?