

**Quizlet project**

**Software Design Specification**

– Hanoi, February 2023 –

Record of changeS

|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
| 28/Feb | A | PhuongNV | * Create UI of folder page * Create new folder and add to database * Delete folder in list folder * CRUD study set in a folder |
| 28/Feb | A | AnhTHV | * Create UI screen home page before login page * Create database * CRUD class * Create UI of class form |
| 28/Feb | A | GiangNH | * Create UI of footer * Search by user name and study set name * Create UI of answers of professors * List answers of professors |
| 28/Feb | M, A | VuBL | * Create login, register * CRUD study sets * Modified animation, UI * Modified CRUD study sets |
| 28/Feb | M, A | HungBM | * Modified a verify, login and register * Modified filter * Create feature your library * Doc * Forget password |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview 4](#_Toc96516286)

[1. Code Packages 4](#_Toc96516287)

[2. Database Design 4](#_Toc96516288)

[a. Database Schema 4](#_Toc96516289)

[b. Table Description 4](#_Toc96516290)

[II. Code Designs 5](#_Toc96516291)

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

[a. Class Diagram 5](#_Toc96516293)

[b. Class Specifications 5](#_Toc96516294)

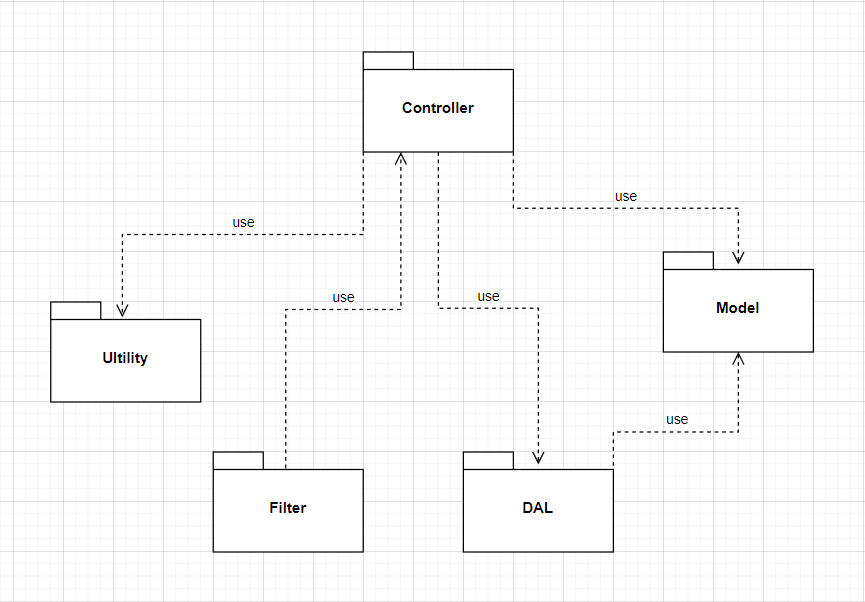
[c. Sequence Diagram(s) 5](#_Toc96516295)

[d. Database queries 6](#_Toc96516296)

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

# I. Overview

## 1. Code Packages



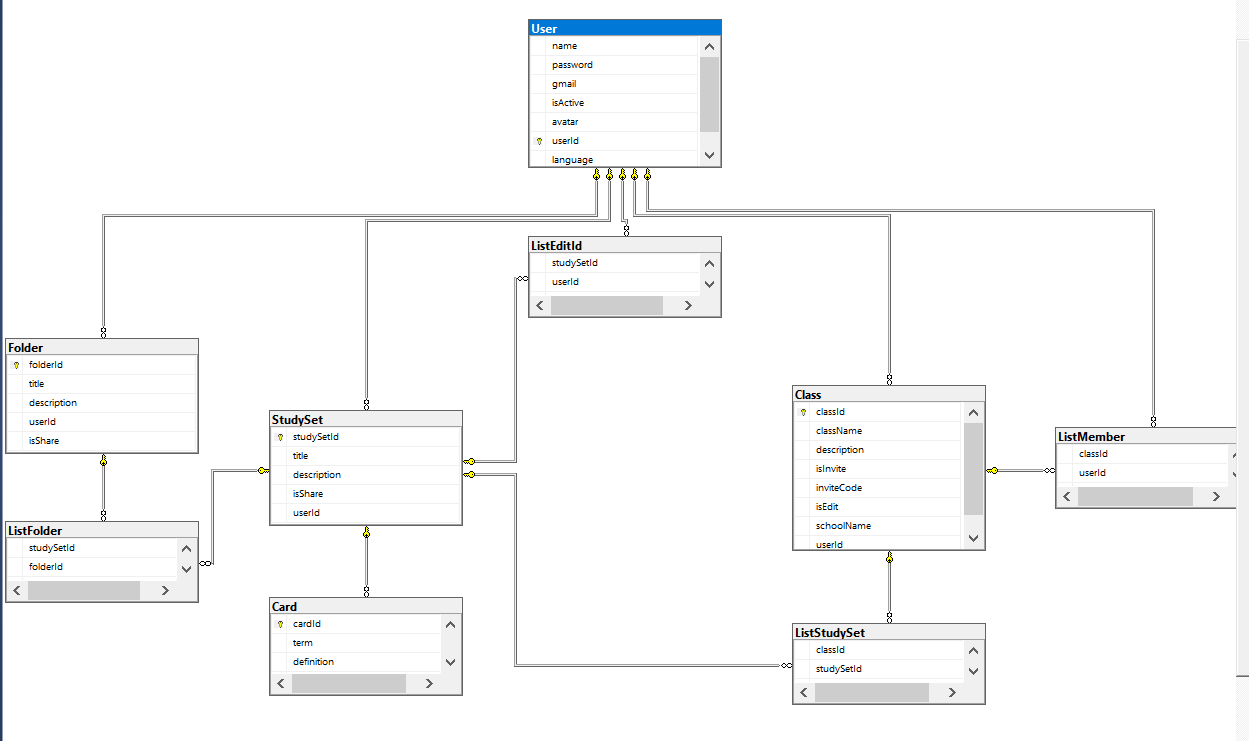
***Package descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *01* | *Controller* | *Controller package included Servlet file* |
| *02* | *Model* | *Model package support represent entities in database* |
| *03* | *DAL* | *DAL package included DBcontext and DAO* |
| *04* | *Ultility* | *Ultility package included some regex and verified file* |
| *05* | *Filter* | *Filter package included filter file to decentralize user and guest* |

## 2. Database Design

### a. Database Schema

*[Provide the tables relationship like example below – following MySQL database naming convention]*



### b. Table Description

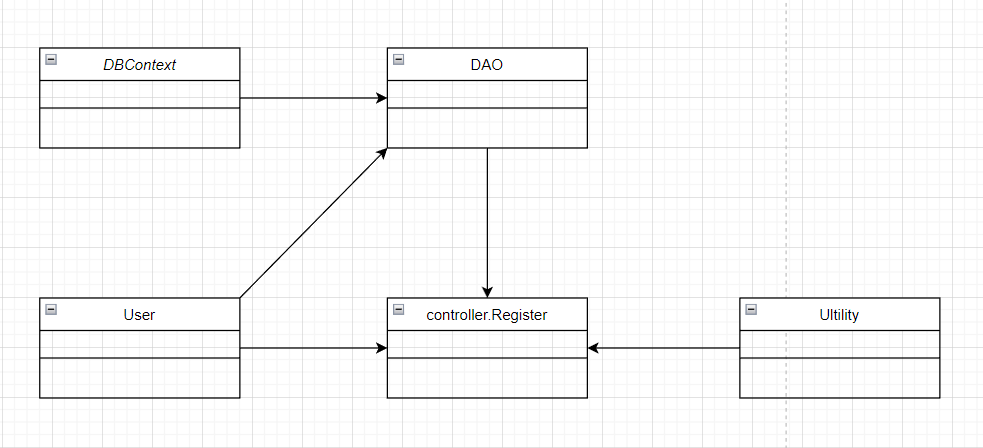
|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| *01* | *User* | *Information of user* |
| *02* | *Folder* | *List information of folder* |
| *03* | *Class* | *List information of class* |
| *04* | *Study set* | *List information of study sets* |
| *05* | *Card* | *List information of cards* |
| *06* | *List Folder* | *List folder of class* |
| *07* | *List Study set* | *List study set in class* |
| *08* | *List Member* | *List members in class* |
| *09* | *List Edit* | *List user to edit study sets* |

# II. Code Designs

## 1. Register

### a. Class Diagram

*[This part presents the class diagram for the relevant feature]*

**

### b. Class Specifications

#### User Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *User(){}* | Default constructor |
| *02* | *User(….){}* | *Parameterized constructor* |
| *03* | *getName(){}* | *Getter of Name* |
| *04* | *setName(string name){}* | *Setter of Name* |
| *05* | *getPass(){}* | *Getter of Pass* |
| *06* | *setPass(string pass){}* | *Setter of Pass* |
| *07* | *getEmail(){}* | *Getter email* |
| *08* | *setEmail(string email){}* | *Setter of email* |
| *09* | *isIsActive(){}* | *Getter of isActive* |
| *10* | *setIsActive(Boolean isActive){}* | *Setter of isActive* |
| *11* | *getAvatar(){}* | *Getter of avatar* |
| *12* | *setAvatar(string avatar)* | *Setter of avatar* |
| *13* | *getLanguage(){}* | *Getter of language* |
| *14* | *setLanguage(string language){}* | *Setter of language* |

#### Ultilities Class

***Class Methods***

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

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *isBlank(string u){}* | *Check string is blank* |
| *02* | *checkExistEmail(string email){}* | *Check existed email* |
| *03* | *ValidInputRequire(string e, string p, string u){}* | *Check valid input* |
| *04* | *validEmail(string e){}* | *Check valid email* |
| *05* | *validUsername(string u){}* | *Check valid username* |
| *06* | *validPassword(string p){}* | *Check valid password* |

#### DAO Class

***Class Methods***

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

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *register(user u){}* | *Insert new user to database* |
|  |  |  |

#### Register Controller Class

***Class Methods***

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

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Doget()* | *Sendredirect to register.jsp* |
| *02* | *doPost()* | *Check input and register* |

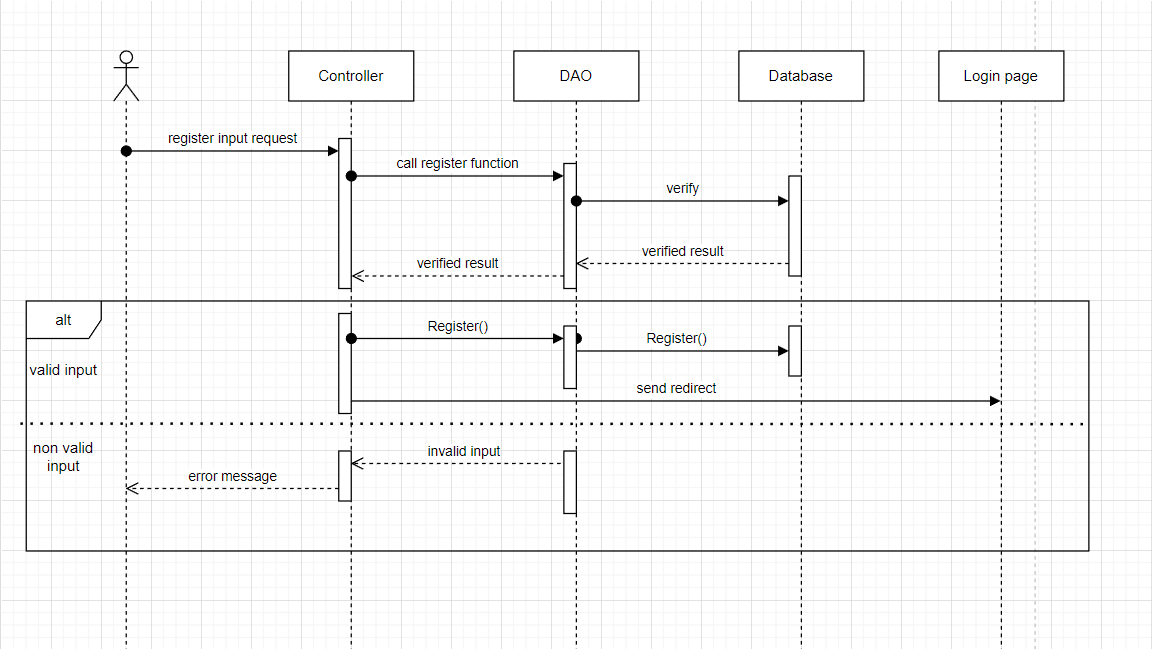
#### DBContext Class

***Class Methods***

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

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *DBContext(){}* | *Connect to database* |

### c. Sequence Diagram(s)



### d. Database Queries

*String sql = "INSERT INTO [dbo].[User]\n"*

*+ " ([name]\n"*

*+ " ,[password]\n"*

*+ " ,[gmail]\n"*

*+ " ,[isActive]\n"*

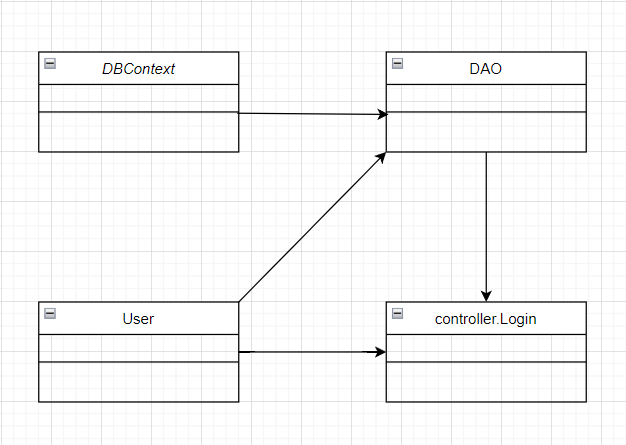
*+ " ,[avatar]\n"*

*+ " ,[language])\n"*

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

## 2. Login

### a. Class Diagram

*[This part presents the class diagram for the relevant feature]* **

### b. Class Specifications

### User Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *User(){}* | Default constructor |
| *02* | *User(….){}* | *Parameterized constructor* |
| *03* | *getName(){}* | *Getter of Name* |
| *04* | *setName(string name){}* | *Setter of Name* |
| *05* | *getPass(){}* | *Getter of Pass* |
| *06* | *setPass(string pass){}* | *Setter of Pass* |
| *07* | *getEmail(){}* | *Getter email* |
| *08* | *setEmail(string email){}* | *Setter of email* |
| *09* | *isIsActive(){}* | *Getter of isActive* |
| *10* | *setIsActive(Boolean isActive){}* | *Setter of isActive* |
| *11* | *getAvatar(){}* | *Getter of avatar* |
| *12* | *setAvatar(string avatar)* | *Setter of avatar* |
| *13* | *getLanguage(){}* | *Getter of language* |
| *14* | *setLanguage(string language){}* | *Setter of language* |

#### DBContext Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *DBContext(){}* | *Connect to database* |

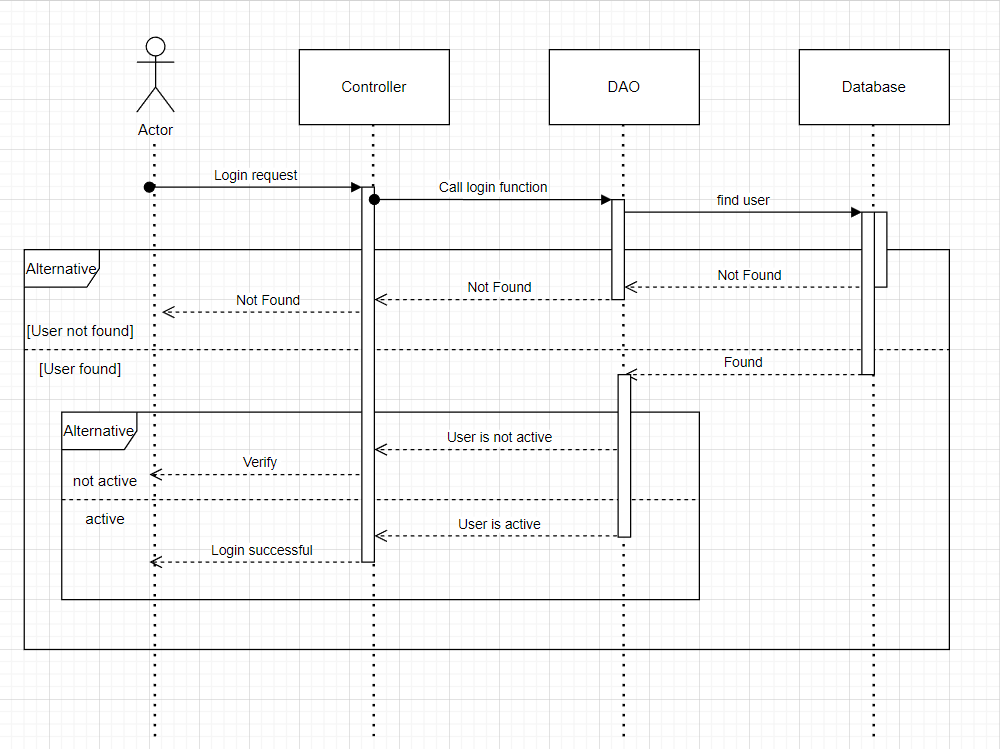
#### Register Controller Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Doget()* | *Get session and Send redirect to login.jsp* |
| *02* | *doPost()* | *Check input and login* |

#### DAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *checkUser(username, password)* | *Check user have username, email, pass to login* |
| *02* | *GetAllUser()* | *Get all user in database* |

### c. Sequence Diagram(s)

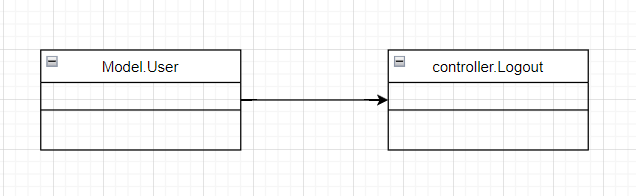


### d. Database Queries

## *select \* from [User]*

## 3. Logout

### a. Class Diagram

**

### b. Class Specifications

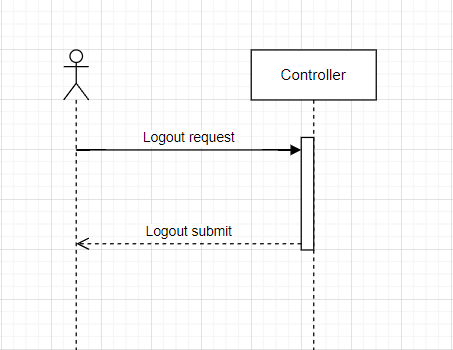
### User Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *User(){}* | Default constructor |
| *02* | *User(….){}* | *Parameterized constructor* |
| *03* | *getName(){}* | *Getter of Name* |
| *04* | *setName(string name){}* | *Setter of Name* |
| *05* | *getPass(){}* | *Getter of Pass* |
| *06* | *setPass(string pass){}* | *Setter of Pass* |
| *07* | *getEmail(){}* | *Getter email* |
| *08* | *setEmail(string email){}* | *Setter of email* |
| *09* | *isIsActive(){}* | *Getter of isActive* |
| *10* | *setIsActive(Boolean isActive){}* | *Setter of isActive* |
| *11* | *getAvatar(){}* | *Getter of avatar* |
| *12* | *setAvatar(string avatar)* | *Setter of avatar* |
| *13* | *getLanguage(){}* | *Getter of language* |
| *14* | *setLanguage(string language){}* | *Setter of language* |

#### Register Controller Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Doget()* | *Remove session to logout* |

### c. Sequence Diagram(s)

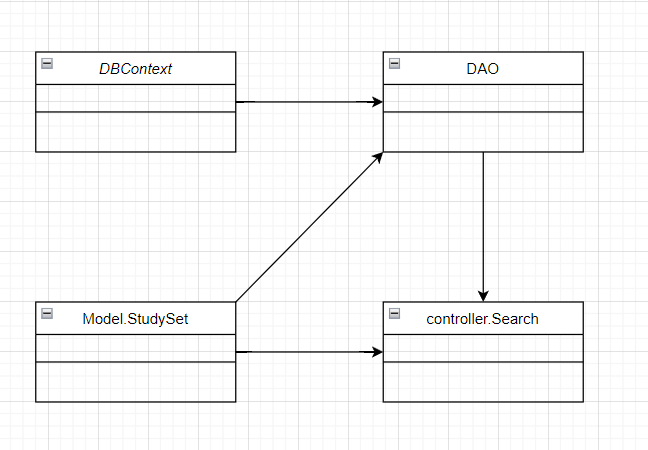


### d. Database Queries

*Logout function do not use database queries.*

## 4. Search study sets by name

### a. Class Diagram

**

### b. Class Specifications

#### Study sets Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *StudySet(){}* | *Default constructor* |
| *02* | *StudySet(….){}* | *Parameterized constructor* |
| *03* | *getId(){}* | *Getter of id* |
| *04* | *setId(int id){}* | *Setter of id* |
| *05* | *getTitle(){}* | *Getter of title* |
| *06* | *setTitle(string title){]* | *Setter of title* |
| *07* | *getDescription(){}* | *Getter of description* |
| *08* | *setDescription(string des){}* | *Setter of description* |
| *09* | *isIsShare(){}* | *Getter of isShare* |
| *10* | *setIsShare(Boolean isShare){]* | *Setter of isShare* |
| *11* | *getuserId(){}* | *Getter of user id* |
| *12* | *setUserid(int id){]* | *Setter of user id* |
| *13* | *getnumberCard(){}* | *Count number of cards in a study set* |
| *14* | *getAuthor(){}* | *Get name of user create study sets* |

#### Search controller Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet(){}* | *Search by name in list study sets* |
|  |  |  |

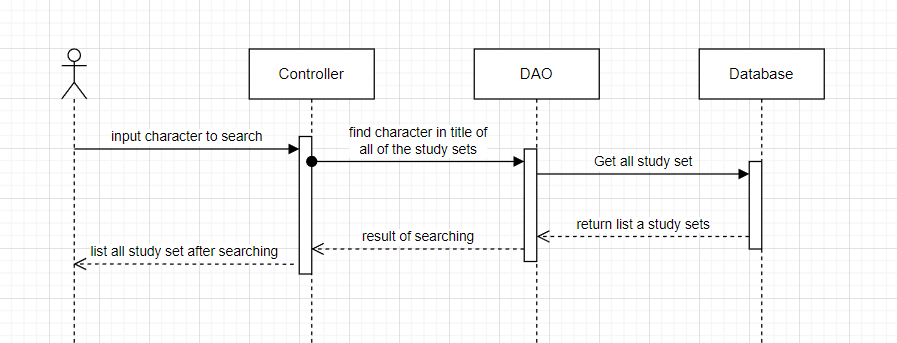
#### DAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *getAllStudySet(){}* | *Get all study sets in database* |
|  |  |  |

#### DBContext Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *DBContext(){}* | *Connect to database* |

### c. Sequence Diagram(s)

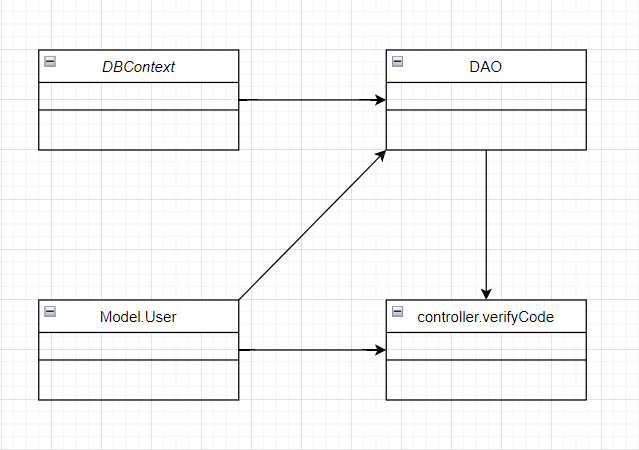


### d. Database Queries

# *Select \* from StudySet*

## 5. Verify code

### a. Class Diagram

**

### b. Class Specifications

### User Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *User(){}* | Default constructor |
| *02* | *User(….){}* | *Parameterized constructor* |
| *03* | *getName(){}* | *Getter of Name* |
| *04* | *setName(string name){}* | *Setter of Name* |
| *05* | *getPass(){}* | *Getter of Pass* |
| *06* | *setPass(string pass){}* | *Setter of Pass* |
| *07* | *getEmail(){}* | *Getter email* |
| *08* | *setEmail(string email){}* | *Setter of email* |
| *09* | *isIsActive(){}* | *Getter of isActive* |
| *10* | *setIsActive(Boolean isActive){}* | *Setter of isActive* |
| *11* | *getAvatar(){}* | *Getter of avatar* |
| *12* | *setAvatar(string avatar)* | *Setter of avatar* |
| *13* | *getLanguage(){}* | *Getter of language* |
| *14* | *setLanguage(string language){}* | *Setter of language* |

#### DBContext Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *DBContext(){}* | *Connect to database* |

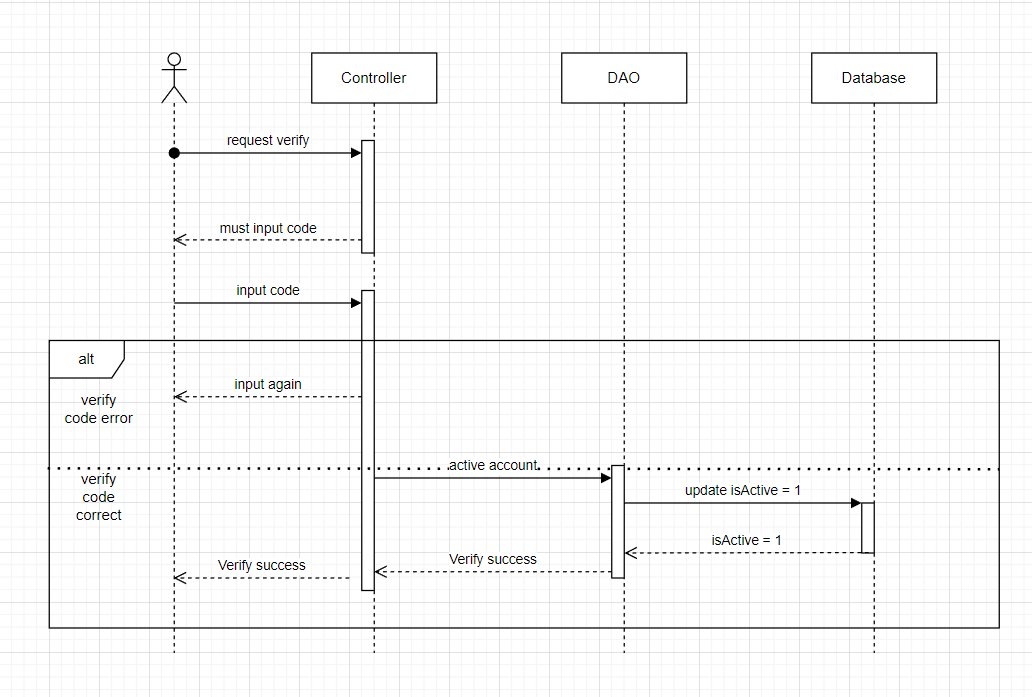
#### Verify controller Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet(){}* | *Send redirect to verify.jsp* |
| *02* | *doPost(){}* | *Active account by verify code* |

#### DAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Active account(string email){}* | *Update is active in database* |

### c. Sequence Diagram(s)

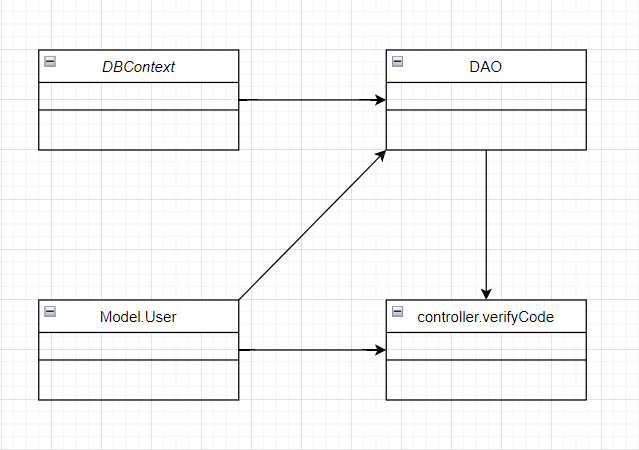


### d. Database Queries

*String sql = "update [User] set isActive = 1 where gmail like ?";*

## 6. Verify code

### a. Class Diagram

**

### b. Class Specifications

### User Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *User(){}* | Default constructor |
| *02* | *User(….){}* | *Parameterized constructor* |
| *03* | *getName(){}* | *Getter of Name* |
| *04* | *setName(string name){}* | *Setter of Name* |
| *05* | *getPass(){}* | *Getter of Pass* |
| *06* | *setPass(string pass){}* | *Setter of Pass* |
| *07* | *getEmail(){}* | *Getter email* |
| *08* | *setEmail(string email){}* | *Setter of email* |
| *09* | *isIsActive(){}* | *Getter of isActive* |
| *10* | *setIsActive(Boolean isActive){}* | *Setter of isActive* |
| *11* | *getAvatar(){}* | *Getter of avatar* |
| *12* | *setAvatar(string avatar)* | *Setter of avatar* |
| *13* | *getLanguage(){}* | *Getter of language* |
| *14* | *setLanguage(string language){}* | *Setter of language* |

#### DBContext Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *DBContext(){}* | *Connect to database* |

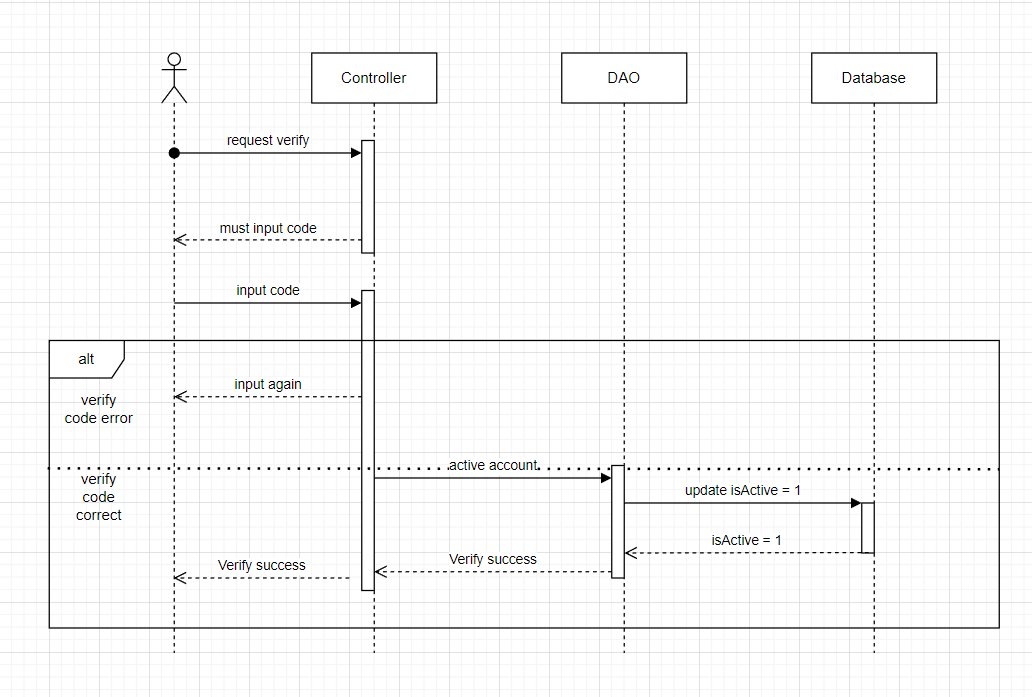
#### Verify controller Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doGet(){}* | *Send redirect to verify.jsp* |
| *02* | *doPost(){}* | *Active account by verify code* |

#### DAO Class

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Active account(string email){}* | *Update is active in database* |

### c. Sequence Diagram(s)



### d. Database Queries

*String sql = "update [User] set isActive = 1 where gmail like ?";*