

**ONLINE LEARNING**

**Software Design Document**

– Hanoi, May 2022 –

**Table of Contents**

[I. Overview](#_heading=h.gjdgxs) **9**

[1. Code Packages/Namespaces](#_heading=h.30j0zll) 9

[2. Coding Conventions](#_heading=h.1fob9te) 10

[II. Code Designs](#_heading=h.3znysh7) **11**

[1. Common/Login](#_heading=h.2et92p0) 11

[a. Class Diagram](#_heading=h.o18eyk5ie0kz) 11

[b. Class Specifications](#_heading=h.g3059os4rjw6) 11

[c. Sequence Diagram(s)](#_heading=h.wp94uy9tbfim) 12

[d. Database queries](#_heading=h.xn4556nezg04) 12

[2. Common/Register](#_heading=h.2s8eyo1) 13

[a. Class Diagram](#_heading=h.49mte0pndtf3) 13

[b. Class Specifications](#_heading=h.9aeboag7dnng) 13

[c. Sequence Diagram(s)](#_heading=h.vm4c7m1yktu2) 14

[d. Database queries](#_heading=h.sr614bnjbbdy) 14

[3. Common/Reset Password](#_heading=h.scy18ihyonru) 15

[a. Class Diagram](#_heading=h.tjk2dtc1knkx) 15

[b. Class Specifications](#_heading=h.iy5o47l23i9r) 15

[c. Sequence Diagram(s)](#_heading=h.edac3mlgxiff) 16

[d. Database queries](#_heading=h.cw3hs3mb16f) 16

[4. Common/User Authorization](#_heading=h.5zkbdid2pz5y) 16

[a. Class Diagram](#_heading=h.t3p0looehzzu) 16

[b. Class Specifications](#_heading=h.8cnul1xakrcx) 17

[c. Sequence Diagram(s)](#_heading=h.9qb9kyauy1i2) 18

[d. Database queries](#_heading=h.qg10816msw82) 18

[5. Common/User Profile](#_heading=h.snru0qwbpzse) 19

[a. Class Diagram](#_heading=h.am8bb1mxs3ya) 19

[b. Class Specifications](#_heading=h.kb4d9pqoulmf) 19

[c. Sequence Diagram(s)](#_heading=h.1tieefn61x0y) 20

[d. Database queries](#_heading=h.gekqyz4qb2te) 20

[6. Common/Change Password](#_heading=h.ltjqt8mgpv11) 21

[a. Class Diagram](#_heading=h.vsici2pjep9q) 21

[b. Class Specifications](#_heading=h.rixbbfyavjbg) 21

[c. Sequence Diagram(s)](#_heading=h.x5dazzbetnzv) 22

[d. Database queries](#_heading=h.wl804jm78so4) 22

[7. Public/Home](#_heading=h.977njpfofmet) 22

[a. Class Diagram](#_heading=h.qp89xxvnh3lq) 22

[b. Class Specifications](#_heading=h.tcjrc0siicwg) 22

[c. Sequence Diagram(s)](#_heading=h.efkdcxbsuc4a) 22

[d. Database queries](#_heading=h.4ug6h4xmeqp6) 23

[8. Public/Blogs List](#_heading=h.hzfn4o2pwxo6) 23

[a. Class Diagram](#_heading=h.bt2owz8x4yfk) 23

[b. Class Specifications](#_heading=h.yiua7ubse5se) 23

[c. Sequence Diagram(s)](#_heading=h.fo0iyzqxvt8k) 24

[d. Database queries](#_heading=h.xh9spgdj3ngb) 24

[9. Public/Blog Details](#_heading=h.ao0w2fsi9kmv) 24

[a. Class Diagram](#_heading=h.ui795srkfxti) 24

[b. Class Specifications](#_heading=h.lqpw7qyu9sdj) 24

[c. Sequence Diagram(s)](#_heading=h.ykptmxpdcb8z) 24

[d. Database queries](#_heading=h.62e27nldkgj) 24

[10. Public/Courses List](#_heading=h.o9wgs08o6q4w) 25

[a. Class Diagram](#_heading=h.r8lytmjy8ze) 25

[b. Class Specifications](#_heading=h.scutwcc7hi4y) 25

[c. Sequence Diagram(s)](#_heading=h.sh848xj5fzx6) 25

[d. Database queries](#_heading=h.kmz210ukdjdf) 25

[11. Public/Course Details](#_heading=h.wa2vh48enzv8) 25

[a. Class Diagram](#_heading=h.d919rnacgv6n) 25

[b. Class Specifications](#_heading=h.xad99psvslej) 25

[c. Sequence Diagram(s)](#_heading=h.yweosztsraab) 25

[d. Database queries](#_heading=h.moutajmwfak) 25

[12. Public/Course Register](#_heading=h.prtis7mylldi) 25

[a. Class Diagram](#_heading=h.tnreirgx9ei4) 25

[b. Class Specifications](#_heading=h.43udly73s52d) 25

[c. Sequence Diagram(s)](#_heading=h.nmi9pax3szv) 25

[d. Database queries](#_heading=h.60hvw6nyjwan) 25

[13. Customer/My Registrations](#_heading=h.t7dedoclf72k) 25

[a. Class Diagram](#_heading=h.vflbm5s4p9ah) 25

[b. Class Specifications](#_heading=h.6sj3rjruphv5) 25

[c. Sequence Diagram(s)](#_heading=h.j6siat2hxbrn) 26

[d. Database queries](#_heading=h.wyrl42mrjz6k) 26

[14. Customer/My Courses](#_heading=h.8n0ta43nppim) 26

[a. Class Diagram](#_heading=h.oc943aad3qst) 26

[b. Class Specifications](#_heading=h.3xiv5ac10u14) 26

[c. Sequence Diagram(s)](#_heading=h.m98mkdr2let9) 26

[d. Database queries](#_heading=h.xwxm4gnliikw) 26

[15. Customer/Lesson View](#_heading=h.4jrjw18n0xes) 26

[a. Class Diagram](#_heading=h.72lpupxyl6uy) 26

[b. Class Specifications](#_heading=h.xu8d7mcg07s0) 26

[c. Sequence Diagram(s)](#_heading=h.v1odyafz7op7) 26

[d. Database queries](#_heading=h.a23u5tnv8vdt) 26

[16. Customer/Quiz Lesson](#_heading=h.l71qqy8z5br4) 26

[a. Class Diagram](#_heading=h.p91wu7un1b28) 26

[b. Class Specifications](#_heading=h.5l2cnpp7skuj) 26

[c. Sequence Diagram(s)](#_heading=h.pwbvn4wd5xrg) 26

[d. Database queries](#_heading=h.wtlnxk64o3c6) 26

[17. Learning/Quiz Handle](#_heading=h.t9huywpfs9el) 27

[a. Class Diagram](#_heading=h.yr01jynvhkf) 27

[b. Class Specifications](#_heading=h.1fncbnfuszgt) 27

[c. Sequence Diagram(s)](#_heading=h.qizgggr9e0hk) 27

[d. Database queries](#_heading=h.scw6e8zgrkzm) 27

[18. Learning/Quiz Review](#_heading=h.hk4l317uvnth) 27

[a. Class Diagram](#_heading=h.v6cxz0buddn) 27

[b. Class Specifications](#_heading=h.j492gscfd4e6) 27

[c. Sequence Diagram(s)](#_heading=h.jxmvm02ff97v) 27

[d. Database queries](#_heading=h.j3at344a8xti) 27

[19. Marketing/Dashboard](#_heading=h.ycdesp6kp7o2) 27

[a. Class Diagram](#_heading=h.egg6ur8r97sh) 27

[b. Class Specifications](#_heading=h.aqakelpmick8) 27

[c. Sequence Diagram(s)](#_heading=h.wj9l1k1q1s0) 27

[d. Database queries](#_heading=h.k7ot644l2s88) 27

[20. Marketing/Post Details](#_heading=h.uh7ke553k1qt) 27

[a. Class Diagram](#_heading=h.1gn65zqkemxa) 27

[b. Class Specifications](#_heading=h.4w3y2p8fe7rf) 27

[c. Sequence Diagram(s)](#_heading=h.d4dbbq1dkjy8) 28

[d. Database queries](#_heading=h.95kukhqd245j) 28

[21. Marketing/Sliders List](#_heading=h.gof38gpeao2p) 28

[a. Class Diagram](#_heading=h.a3q75w2pxslq) 28

[b. Class Specifications](#_heading=h.mm8lrjmmtze3) 28

[c. Sequence Diagram(s)](#_heading=h.fw6q3pu49vg0) 28

[d. Database queries](#_heading=h.o1m2tto6g0x3) 28

[22. Marketing/Slider Details](#_heading=h.ocqddnc7oolt) 28

[a. Class Diagram](#_heading=h.a8xh078p92gy) 28

[b. Class Specifications](#_heading=h.hacvdoe1ydwh) 28

[c. Sequence Diagram(s)](#_heading=h.3uj1lhjv1a9j) 28

[d. Database queries](#_heading=h.t76qohy24iua) 28

[23. Course Content/Subjects List](#_heading=h.3qx2wn1s630f) 28

[a. Class Diagram](#_heading=h.xed8cui074b3) 28

[b. Class Specifications](#_heading=h.enaea77v62o3) 28

[c. Sequence Diagram(s)](#_heading=h.nel0z29raubp) 28

[d. Database queries](#_heading=h.vqhvl8eidjcw) 28

[24. Course Content/New Subject](#_heading=h.gh0hclosnw37) 28

[a. Class Diagram](#_heading=h.e3kn05cb6jmx) 28

[b. Class Specifications](#_heading=h.hhoy388e4owr) 29

[c. Sequence Diagram(s)](#_heading=h.sgmbtkgycxai) 29

[d. Database queries](#_heading=h.mrnu84k4v77t) 29

[25. Course Content/Subject Details](#_heading=h.ucw66ujetv0a) 29

[a. Class Diagram](#_heading=h.7erdj9w0ryjp) 29

[b. Class Specifications](#_heading=h.3s4wtb77bwni) 29

[c. Sequence Diagram(s)](#_heading=h.nemixjj52a4y) 29

[d. Database queries](#_heading=h.5339humo9djf) 29

[26. Course Content/Subject Dimension](#_heading=h.aqw4w1r4gmv4) 29

[a. Class Diagram](#_heading=h.4v1w5rllxsfy) 29

[b. Class Specifications](#_heading=h.20jrz5rbm5i5) 29

[c. Sequence Diagram(s)](#_heading=h.b7hr9pzo4v3) 29

[d. Database queries](#_heading=h.mz8085yf0kso) 29

[27. Course Content/Price Package](#_heading=h.nmc3uk7nz2ua) 29

[a. Class Diagram](#_heading=h.pbx6uihuqbq5) 29

[b. Class Specifications](#_heading=h.do2n43w55gkp) 29

[c. Sequence Diagram(s)](#_heading=h.262cf1f34f54) 29

[d. Database queries](#_heading=h.3m25ymtktoya) 29

[28. Course Content/Subject Lessons](#_heading=h.kidv5ruapoan) 30

[a. Class Diagram](#_heading=h.h5lq94g5483n) 30

[b. Class Specifications](#_heading=h.7e988dp8sx4b) 30

[c. Sequence Diagram(s)](#_heading=h.e4wmyacdl3o6) 30

[d. Database queries](#_heading=h.ndfxrofrjim) 30

[29. Course Content/Lesson Details](#_heading=h.cs38mvexf48f) 30

[a. Class Diagram](#_heading=h.yyt50b95ya10) 30

[b. Class Specifications](#_heading=h.hrikfd5qzu5a) 30

[c. Sequence Diagram(s)](#_heading=h.ke98fhfpwwou) 30

[d. Database queries](#_heading=h.bimi5iebkvlk) 30

[30. Test Content/Questions List](#_heading=h.1br3sqcefll3) 30

[a. Class Diagram](#_heading=h.gqkhbxbudum5) 30

[b. Class Specifications](#_heading=h.nblcnmqe20zf) 30

[c. Sequence Diagram(s)](#_heading=h.n4ikx3m5hcxr) 30

[d. Database queries](#_heading=h.9ldnzftdkhtf) 30

[31. Test Content/Questions Details](#_heading=h.uecevw4t73tz) 30

[a. Class Diagram](#_heading=h.tfmzdhb1mplr) 30

[b. Class Specifications](#_heading=h.brns1430kum3) 31

[c. Sequence Diagram(s)](#_heading=h.sifbrv5jf5bc) 31

[d. Database queries](#_heading=h.715qfjf7dfz5) 31

[32. Test Content/Questions Import](#_heading=h.hkxaa81jmpcv) 31

[a. Class Diagram](#_heading=h.1vyw8fvxht2h) 31

[b. Class Specifications](#_heading=h.uanbgde1nbgf) 31

[c. Sequence Diagram(s)](#_heading=h.ka1xbug5iyxo) 31

[d. Database queries](#_heading=h.ue7qwa4d09jl) 31

[33. Test Content/ Quizzes List](#_heading=h.w56imzkys5y4) 31

[a. Class Diagram](#_heading=h.r6o2n8dnm18d) 31

[b. Class Specifications](#_heading=h.tfhj5c7kmyhe) 31

[c. Sequence Diagram(s)](#_heading=h.lys0q3v6evqp) 31

[d. Database queries](#_heading=h.ucwtwzomhfj) 31

[34. Test Content/ Quizzes Details](#_heading=h.exyz5mtxd0n9) 31

[a. Class Diagram](#_heading=h.v31ksrks4nqk) 31

[b. Class Specifications](#_heading=h.t3od62jhr4j2) 31

[c. Sequence Diagram(s)](#_heading=h.s86o1tqb6hov) 32

[d. Database queries](#_heading=h.3qq27vhpc011) 32

[35. Sale/ Registrations List](#_heading=h.gffqok5xrn7r) 32

[a. Class Diagram](#_heading=h.uv6pgcezdo6) 32

[b. Class Specifications](#_heading=h.y3235pu0kp0i) 32

[c. Sequence Diagram(s)](#_heading=h.5bwsm4889lar) 32

[d. Database queries](#_heading=h.ws23hi3tejob) 32

[36. Sale/ Registrations Details](#_heading=h.wtagkn4qaz57) 32

[a. Class Diagram](#_heading=h.sm7qb2t2xtw) 32

[b. Class Specifications](#_heading=h.7c4lk8h3z3vj) 32

[c. Sequence Diagram(s)](#_heading=h.d15dqdv3smhf) 32

[d. Database queries](#_heading=h.xrm4k9x6d6ui) 32

[37. Admin/User List](#_heading=h.epfiwjs6vqj7) 32

[a. Class Diagram](#_heading=h.byl5lpi4ow6l) 32

[b. Class Specifications](#_heading=h.jbfm2w7tagvb) 32

[c. Sequence Diagram(s)](#_heading=h.hd66mxho7w18) 32

[d. Database queries](#_heading=h.wm9ida95ov0v) 32

[38. Admin/User Details](#_heading=h.r7onn679ftt7) 33

[a. Class Diagram](#_heading=h.kk0uzud9m94x) 33

[b. Class Specifications](#_heading=h.au0g2aivfahd) 33

[c. Sequence Diagram(s)](#_heading=h.10182e2fnzdi) 33

[d. Database queries](#_heading=h.n9q6w6x5qjv7) 33

[39. Admin/Settings List](#_heading=h.d2vc5aosvl6) 34

[a. Class Diagram](#_heading=h.7f9uwa2isk0k) 34

[b. Class Specifications](#_heading=h.8zyum7o2fp77) 34

[c. Sequence Diagram(s)](#_heading=h.5zwettmzawu7) 35

[d. Database queries](#_heading=h.nhg3cgk5ohmk) 36

[40. Admin/Settings Details](#_heading=h.w5bn9bt6gtt5) 37

[a. Class Diagram](#_heading=h.2cw9vj30gqq2) 37

[b. Class Specifications](#_heading=h.39zs1xb7dojw) 38

[c. Sequence Diagram(s)](#_heading=h.p0tw2pnqvv9n) 38

[d. Database queries](#_heading=h.gjdvblr8msej) 38

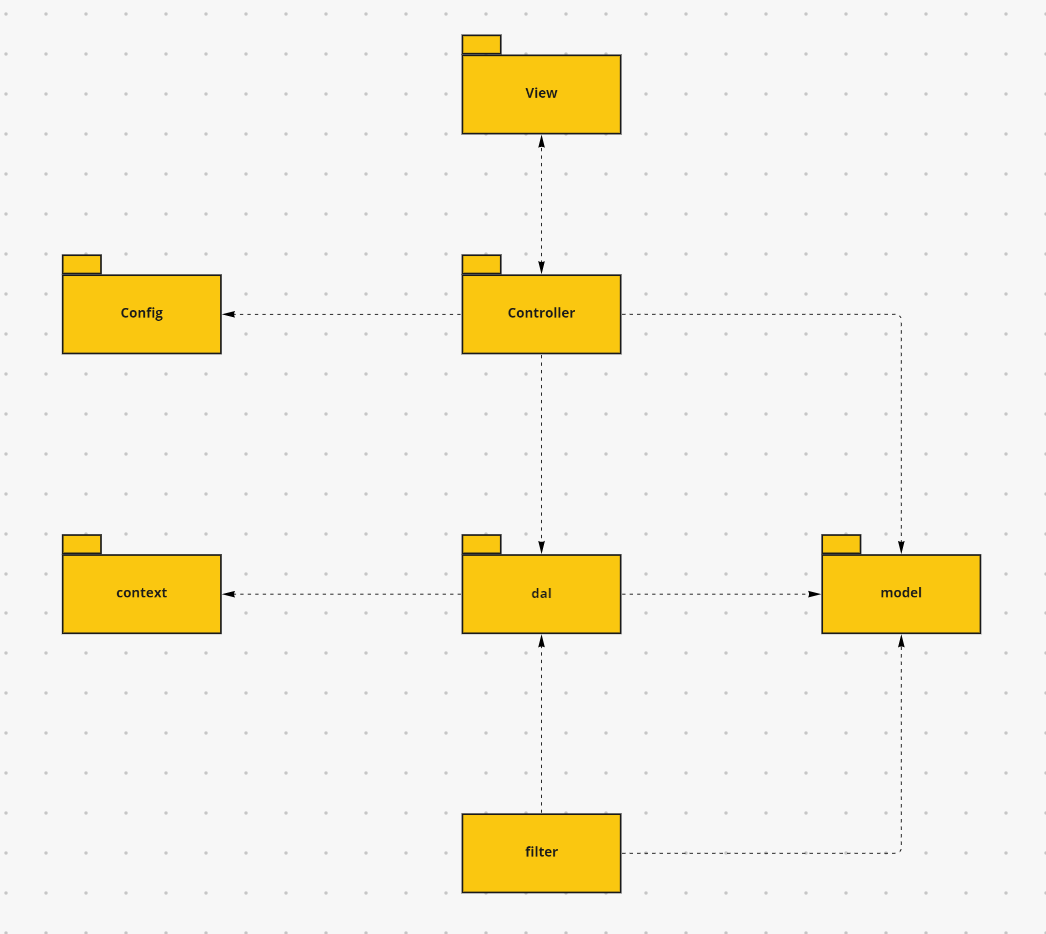
[III. Database Design](#_heading=h.17dp8vu) **38**

[1. Database Schema](#_heading=h.3rdcrjn) 38

[2. Table Description](#_heading=h.26in1rg) 38

# I. Overview

## 1. Code Packages/Namespaces



***Package descriptions & package class naming conventions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *config* | *Decrepitation: This component helps define configuration*  *Naming convention: variables inside classes must be upper case – snake case.* |
| *02* | *dal* | *Decrepitation: This package contains all the classes that can interact with the database.*  *Naming convention: class name must have “DBContext” at the end* |
| *03* | *controller* | *Decrepitation: This package contains controllers which enable the interaction between views and models.*  *Naming convention: class name must have “Controller” at the end* |
| *04* | *context* | *Decrepitation: This component helps connect database* |
| *05* | *model* | *Decrepitation: This component includes interfaces used to receive data retrieved from the database and classes used to receive data* |
| *06* | *view* | *Decrepitation: This component displays front-end.* |
| *07* | *filter* | *Decrepitation: This component authentication and access uri* |

*Note about Name convention: All classes in the package have the key ”controller” must have “controller” at the end.*

## 2. Coding Conventions

General naming conventions (variable names, constant names, methods, classes, objects, etc.):

- Write in English.

- Correct spelling.

- Meaningful, true to the functional characteristics of the corresponding object.

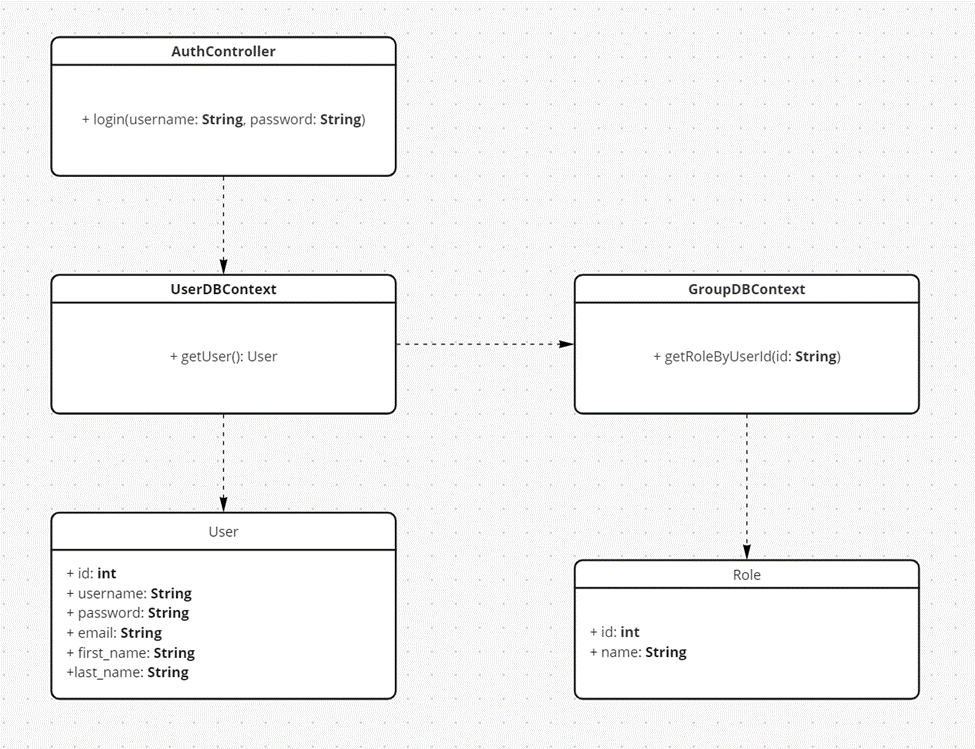
Regulations on naming conventions:

| **Indentifier** | Rules for Naming | **Example** |
| --- | --- | --- |
| Classes | Class names should be nouns, in mixed case with the first letter of each internal word capo-tailed. Try to keep your class names simple and descriptive. Use whole words avoid acronyms and abbreviations (unless the abbreviation is much more widely used than the long form, such as URL or HTML) | class Account;  class CategorySubject; |
| Interfaces | Interface names should be capitalized like class names | interface Storing;  interface RasterDelegate; |
| Methods | Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized | checkAccountIsExist() |
| Variables | Except for variables, all instance, class and class constraints are in mixed case with a lower-case first letter. Internal words start with capital letters.  Variable names should be short yet meaning-full. The choice of a variable name should be mnemonic-that is, designed to indicate to the casual observer the intent of its user. One-char-acter variable names should be avoided except for temporary “throwaway” variables. Com-mom names for temporary variables are i, j, k, m and n for integers; c, d and e for characters. | int number;  String name; |
| Constants | The names of variables declared class con-stands and of ANSI constraints should be all uppercase with words separated by under-scores (“\_”). (ANSI constraints should be avoided, for ease of debugging) | Int MIN\_WIDTH = 4; |

# II. Code Designs

## 1. Common/Login

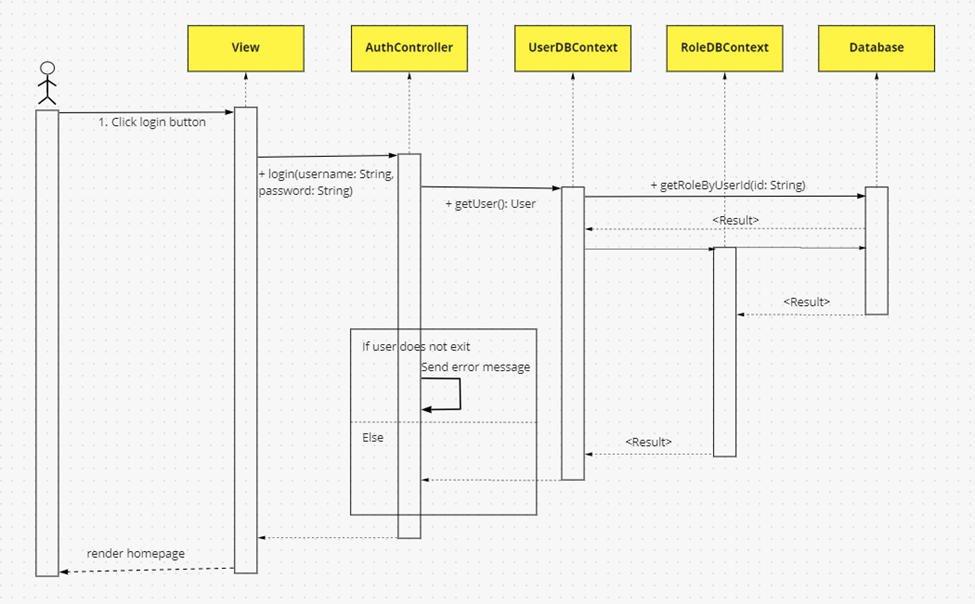
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Login | Using to login into the system |
| 02 | getUser | Using to check user in database |
| 03 | getRoleByUserId | Using get role of user in database |

### c. Sequence Diagram(s)



### d. Database queries

**String sql = "SELECT \* FROM users where username=? AND password=?"**

## 2. Common/Register

### a. Class Diagram

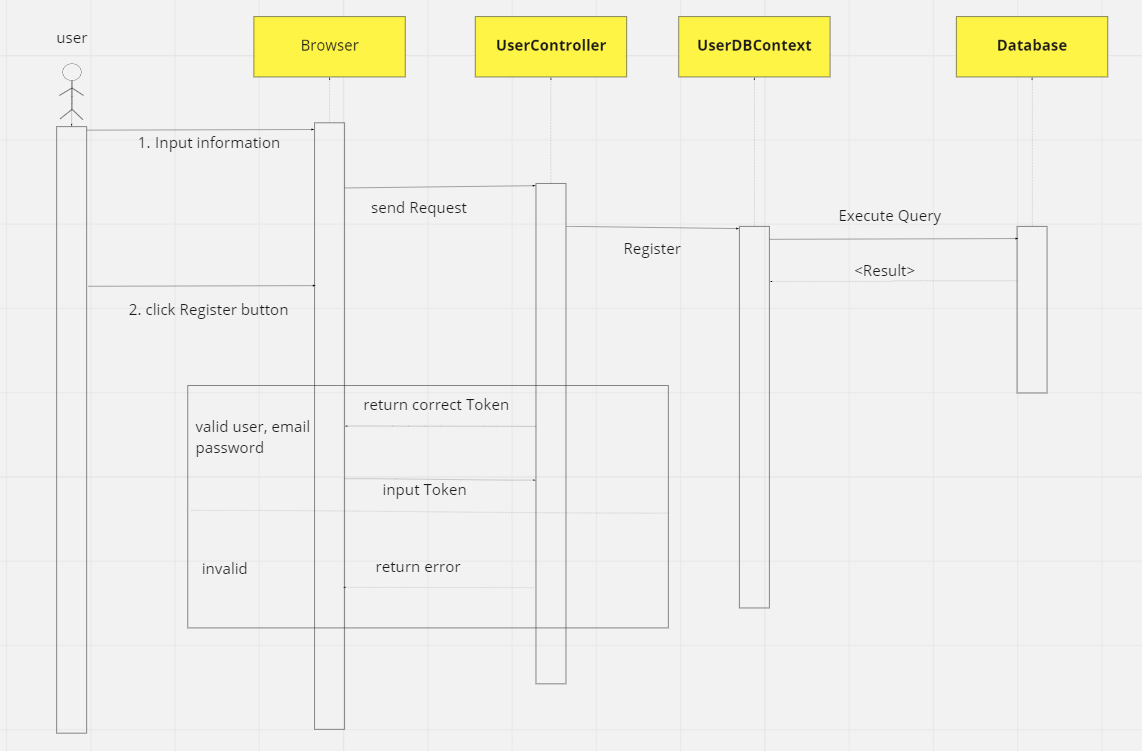


### b. Class Specifications

#### 

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 1 | **SignUp** | **create new Account** |
| 2 | **checkAccount** | **check username or email exist** |
| 3 | **insert** | **insert data (new account) into database** |
| 4 | **sendMail** | **send mail to user** |

### c. Sequence Diagram(s)



### d. Database queries

**INSERT INTO [user]**

**([username], [password], [email], [phone], [first\_name], [last\_name], [gender], [avatar], [is\_staff], [is\_super], [enable], [is\_active])**

**VALUES**

**(?,?,?,?,?,?,?,?,?,?,?,?)**

## 3. Common/Reset Password

### a. Class Diagram

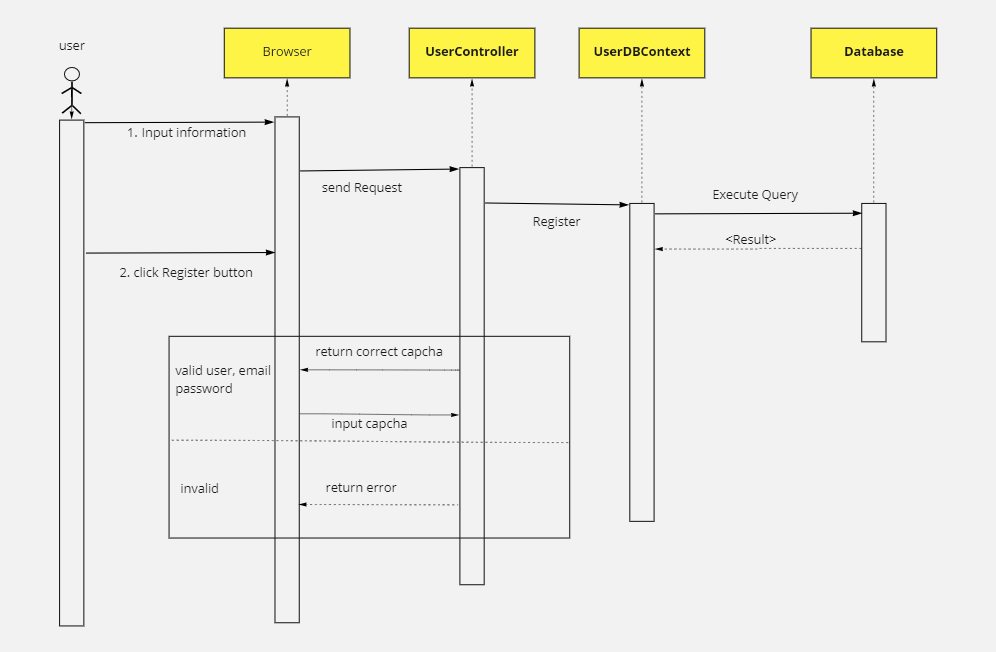
### 

### b. Class Specifications

#### 

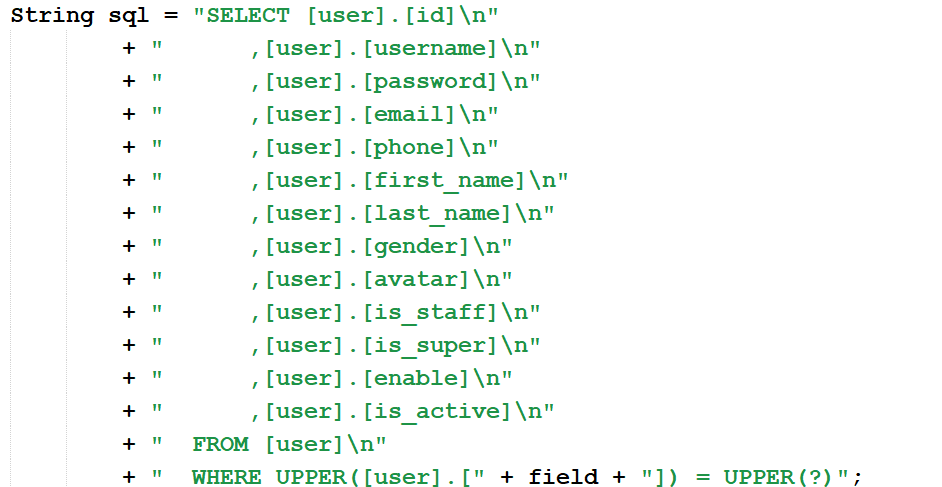
| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **changePassword** | **using to change password** |
| **2** | **checkPassword** | **using to check validate password** |
| **3** | **enCode** | **using to encode password** |
| **4** | **getPassword** | **using to update password** |
| **5** | **recover** | **using to save password** |

### c. Sequence Diagram(s)

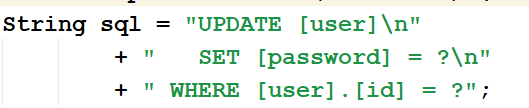


### d. Database queries

#### Get user from database:

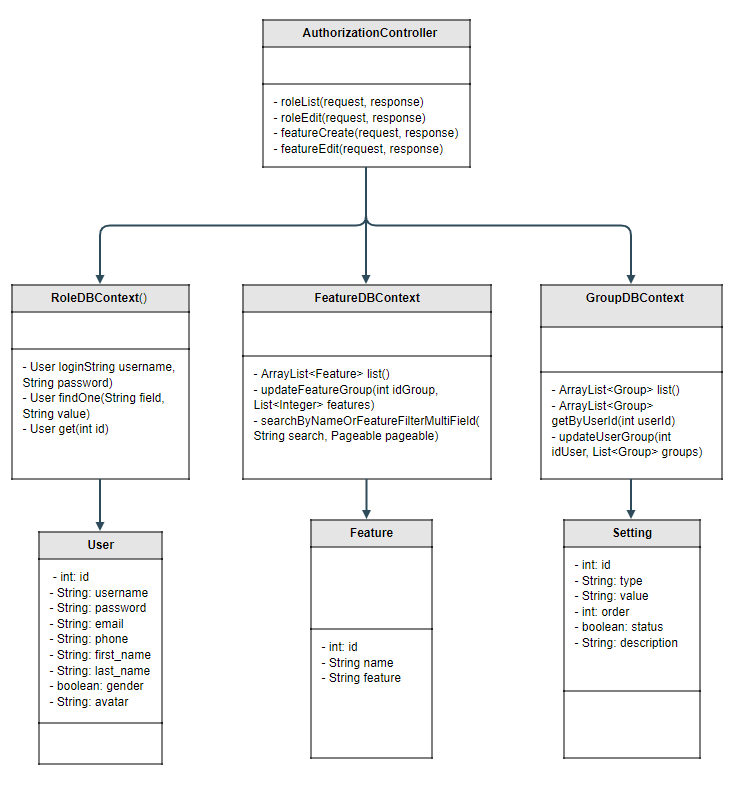
****

#### Change password from database:

****

## 4. Common/User Authorization

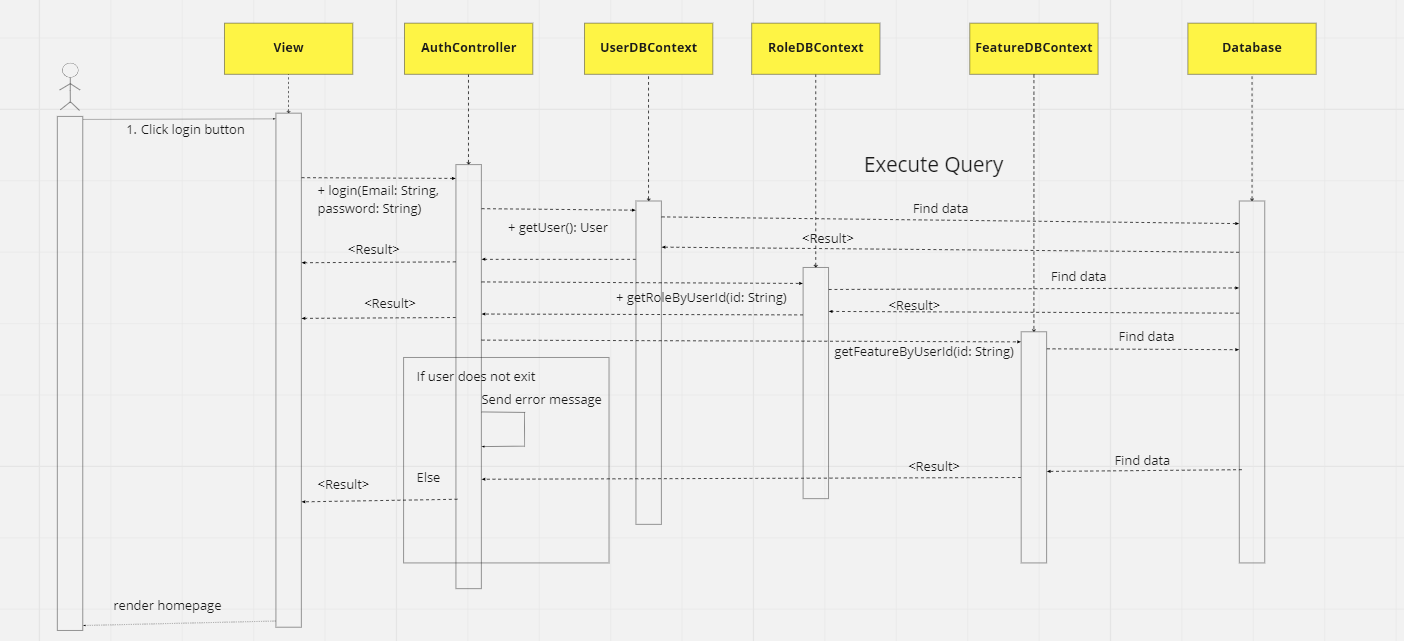
### a. Class Diagram



### b. Class Specifications

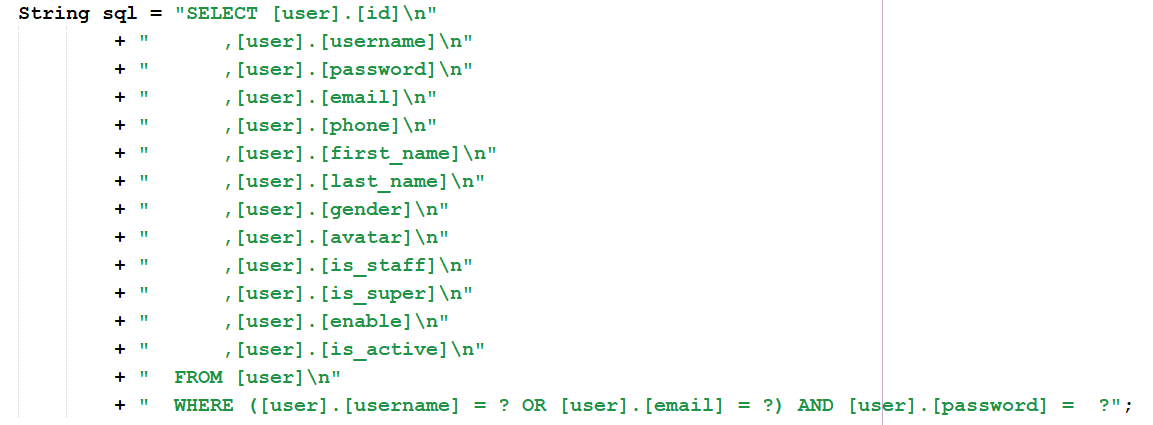
| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **login** | **Using to login into the system** |
| **2** | **findOne** | **get user by field** |
| **3** | **listFeature** | **get all user by feature** |
| **4** | **searchByNameOrFeatureFilterMultiField** | **search and filter user by field** |
| **5** | **updateFeatureGroup** | **update user feature** |

### c. Sequence Diagram(s)

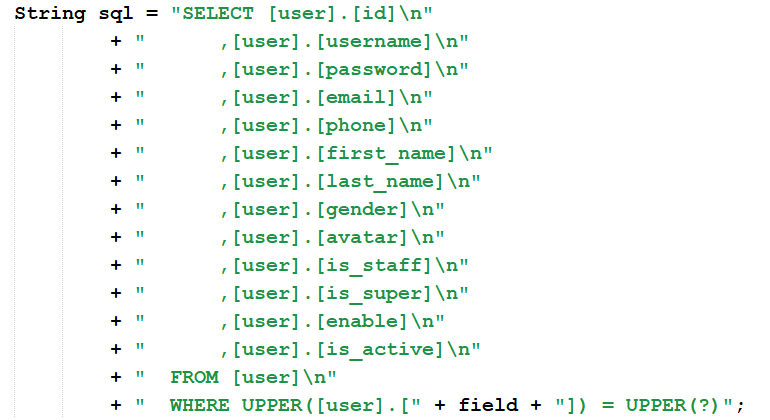


### d. Database queries

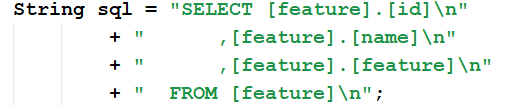
#### Login:



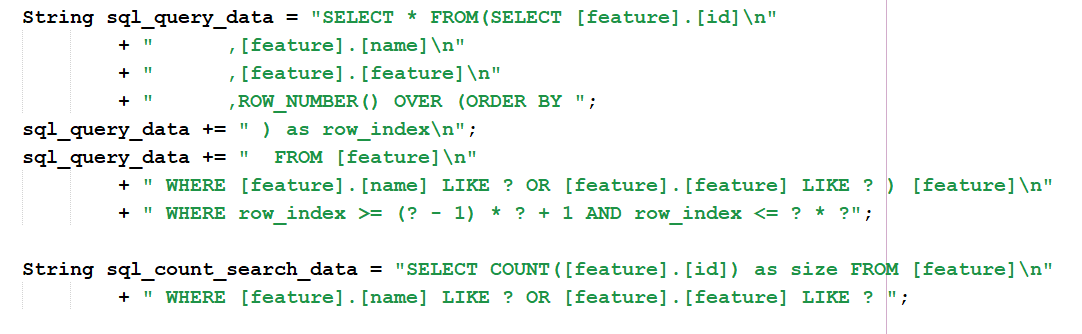
FindOne:



listFeature:

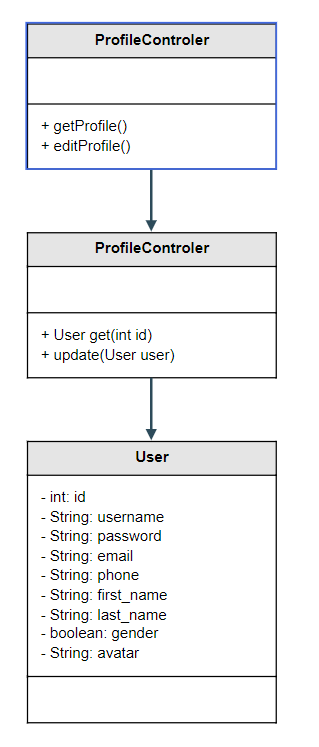


searchByNameOrFeatureFilterMultiField:



## 5. Common/User Profile

### a. Class Diagram

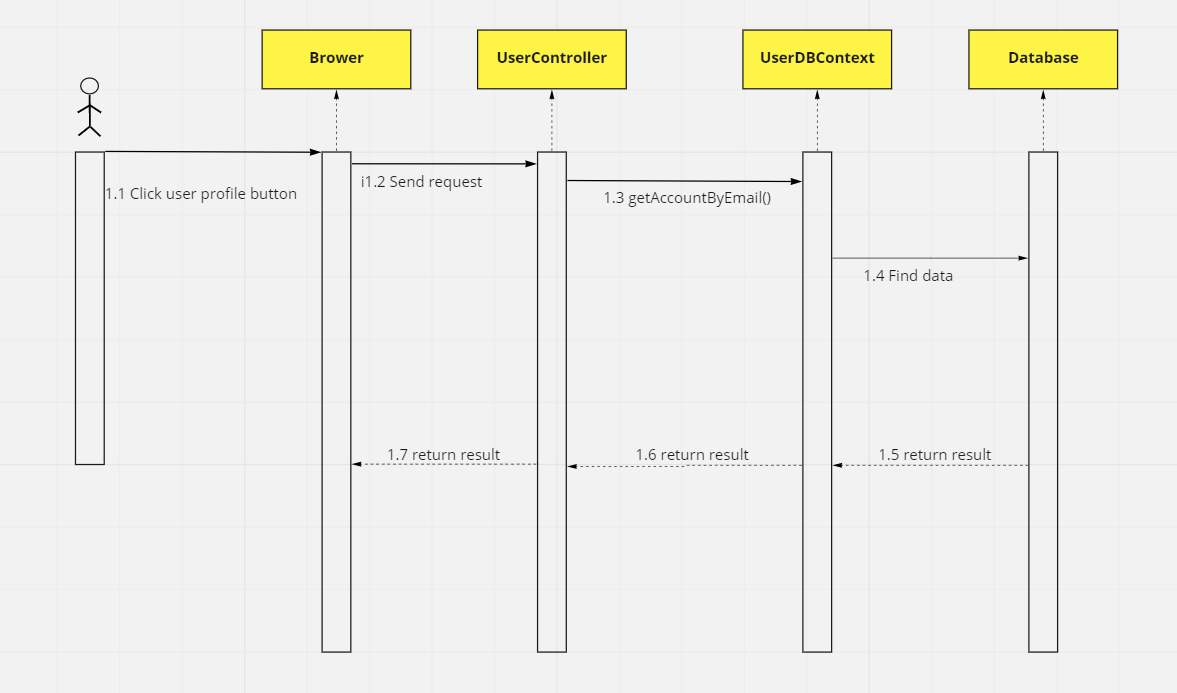


### b. Class Specifications

#### 

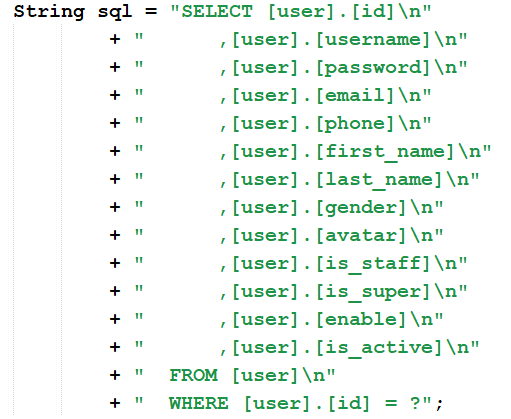
| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getUserById** | **use to get account profile by id** |
| **2** | **updateProfile** | **use to update user’s profile** |

### c. Sequence Diagram(s)

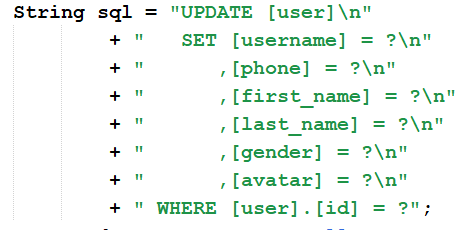


### d. Database queries

* **getUserById:**

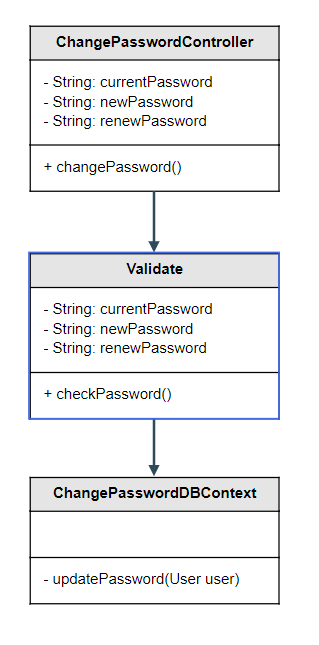
****

* **updateProfile:**

****

## 6. Common/Change Password

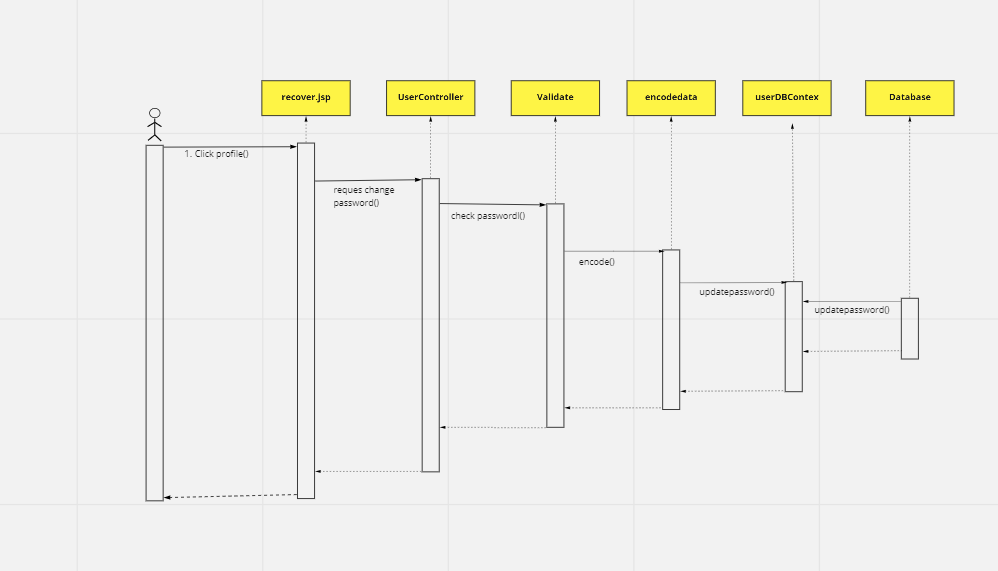
### a. Class Diagram



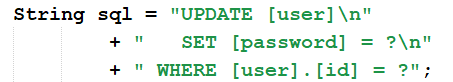
### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **changePassword** | **using to change password** |
| **2** | **checkPassword** | **using to check validate password** |
| **3** | **updatePassword** | **using to update password** |

### c. Sequence Diagram(s)



### d. Database queries



## 7. Public/Home

### a. Class Diagram

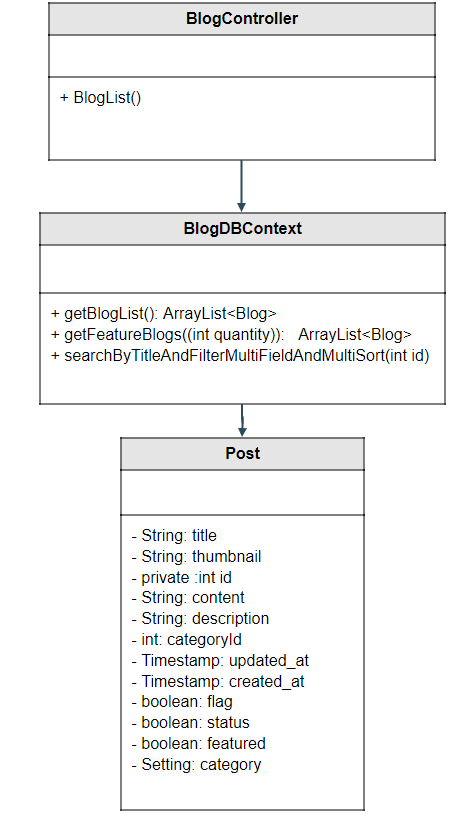
### b. Class Specifications

### c. Sequence Diagram(s)

### d. Database queries

## 8. Public/Blogs List

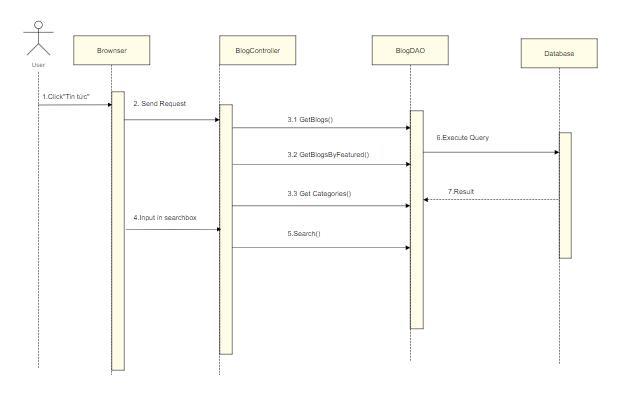
### a. Class Diagram

****

### b. Class Specifications

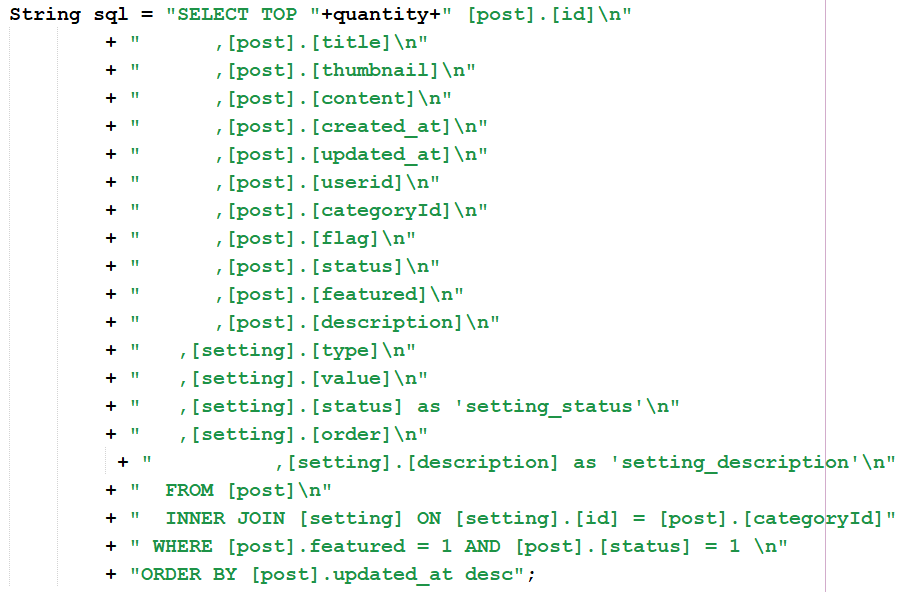
| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getBlogList** | **get all blog pagination** |
| **2** | **getFeatureBlogs** | **get blog which featured** |
| **3** | **searchByTitleAndFiltetMultiFieldAndMultiSort()** | * **get all Blog in database** * **User can search Blog by title** * **User can filter by Category** |

### c. Sequence Diagram(s)

****

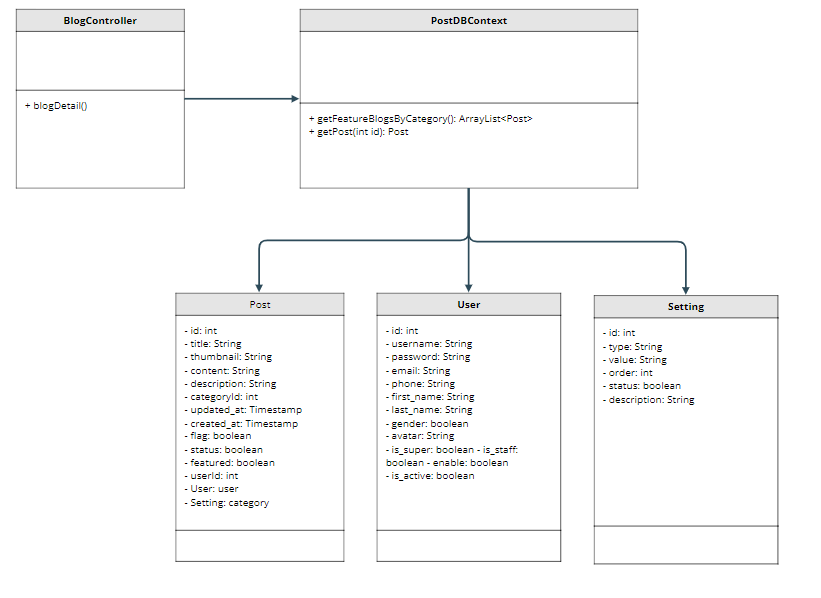
### d. Database queries

* **getFeatureBlogs:**

****

## 9. Public/Blog Details

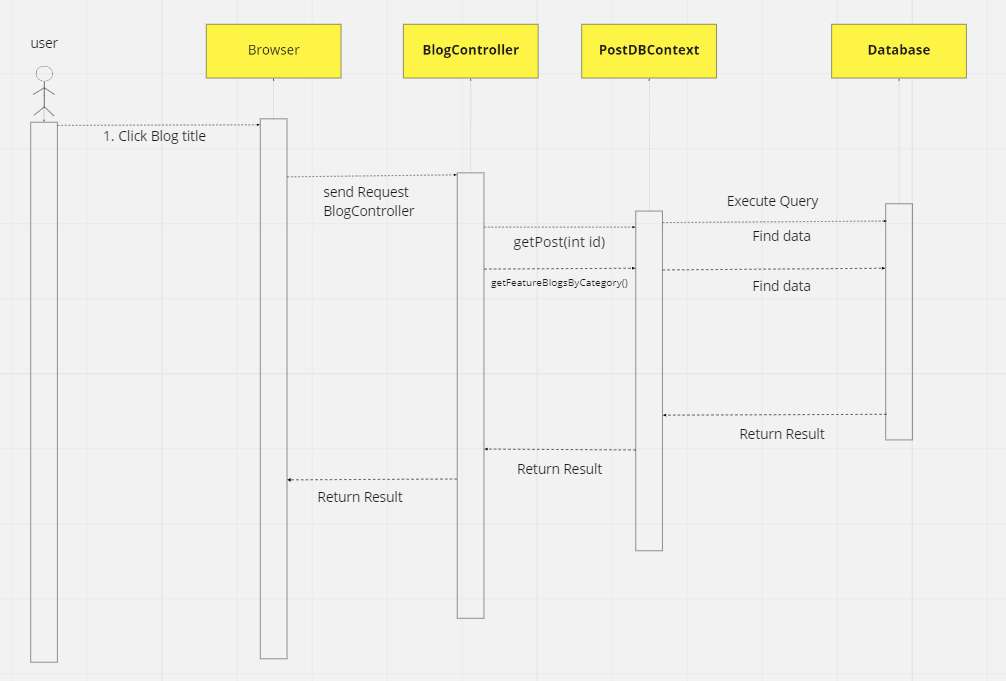
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getFeatureBlogsByCategory()** | **get random articles with the same category and featured** |
| **2** | **getPost()** | **get 1 blog by id** |

### c. Sequence Diagram(s)



### d. Database queries

* **SELECT \***

**FROM [post] INNER JOIN [setting] ON [setting].[id] = [post].[categoryId]**

**INNER JOIN [user] ON [user].[id] = [post].[userid]**

**WHERE [post].[id] = ?**

* **SELECT \***

**FROM [post] INNER JOIN [setting] ON [setting].[id] = [post].[categoryId]**

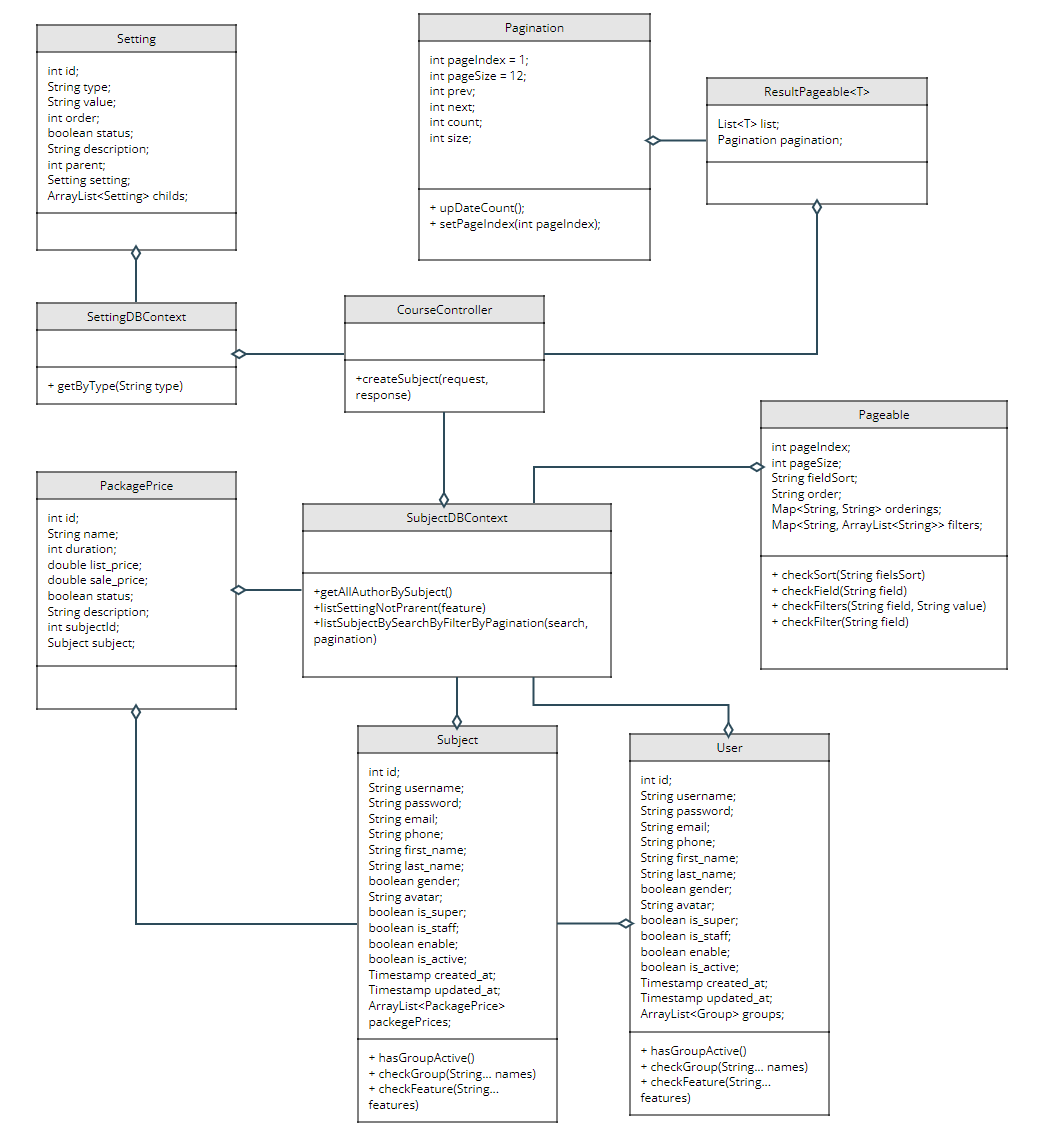
**INNER JOIN [user] ON [user].[id] = [post].[userid]**

**WHERE [post].[featured] = 1 AND [post].[categoryId] = ? AND [post].[status] = 1**

**ORDER BY NEWID()**

## 10. Public/Courses List

### a. Class Diagram



### b. Class Specifications

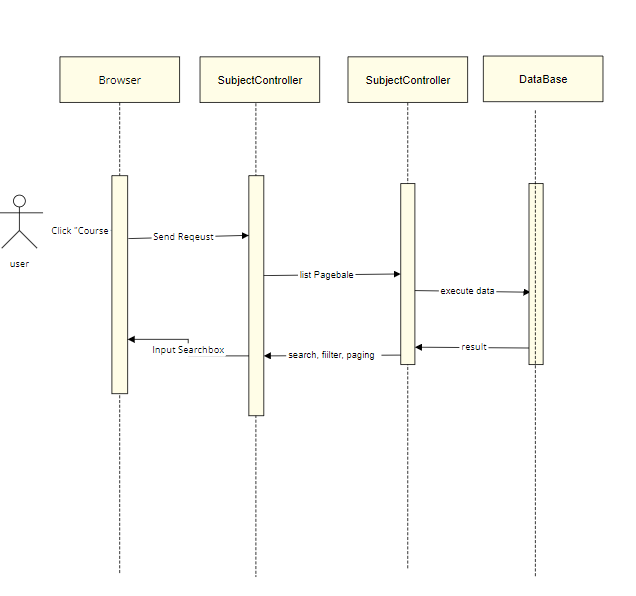
**Class CourseController**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **listCourse** | **do get list course from server** |

**Class SubjectDBContext**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getAllAuthorBySubject()** | **get all the authors of the courses that have been published for the filter from the author** |
| **2** | **getSettingNotParent()** | **get all the categories of published courses for filter from the categories** |
| **3** | **listSubjectBySearchByFilterByPagination(search, pagination)** | **get all courses by search, filter, pagination** |

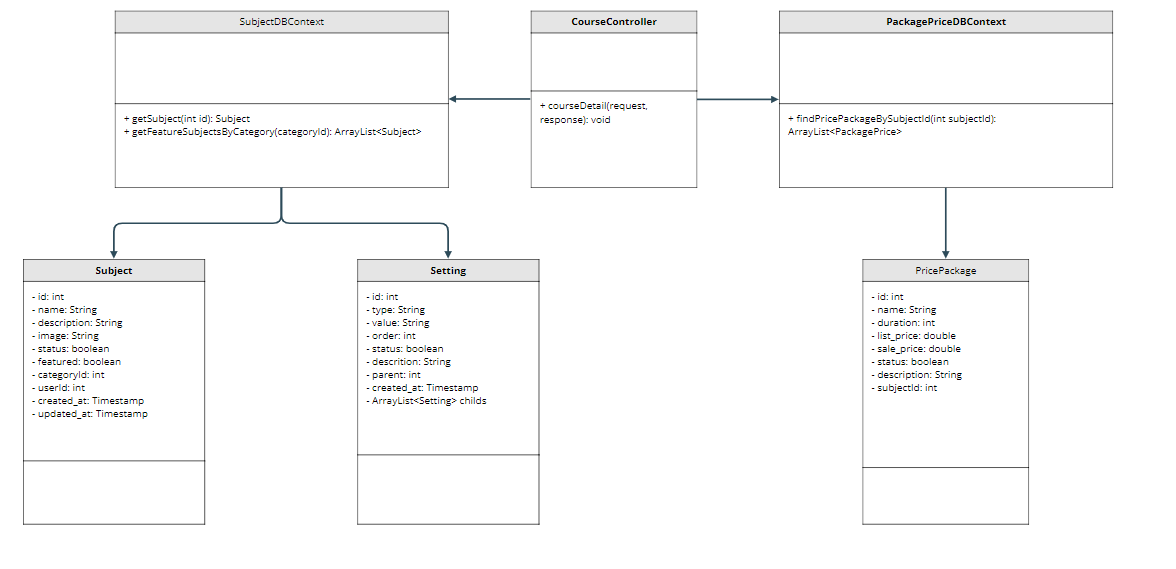
### c. Sequence Diagram(s)



### d. Database queries

## 11. Public/Course Details

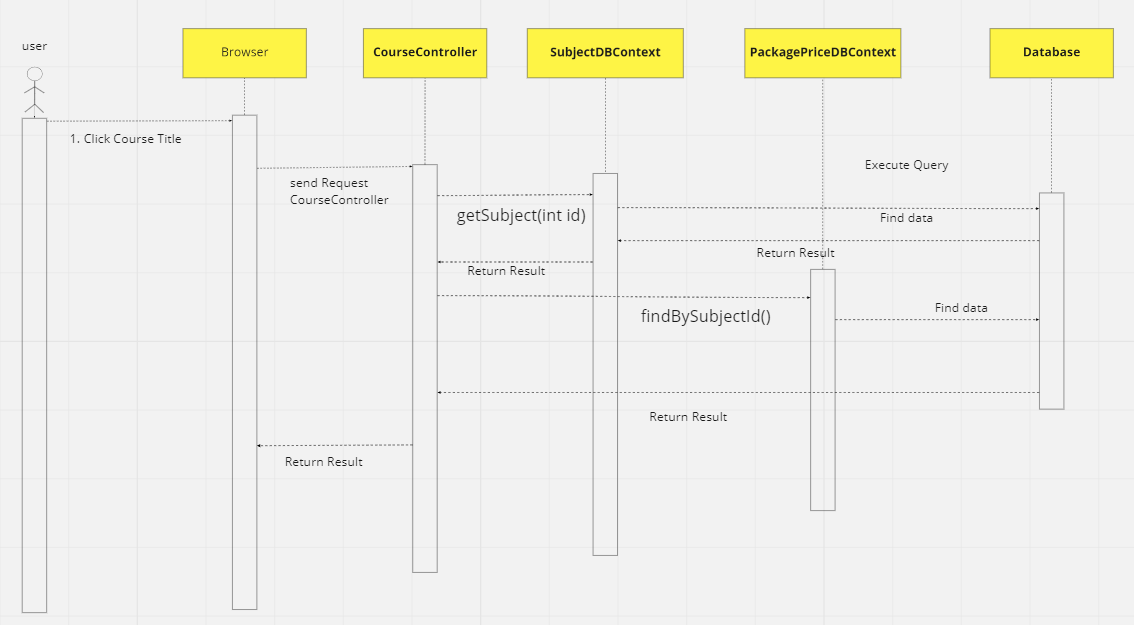
### a. Class Diagram



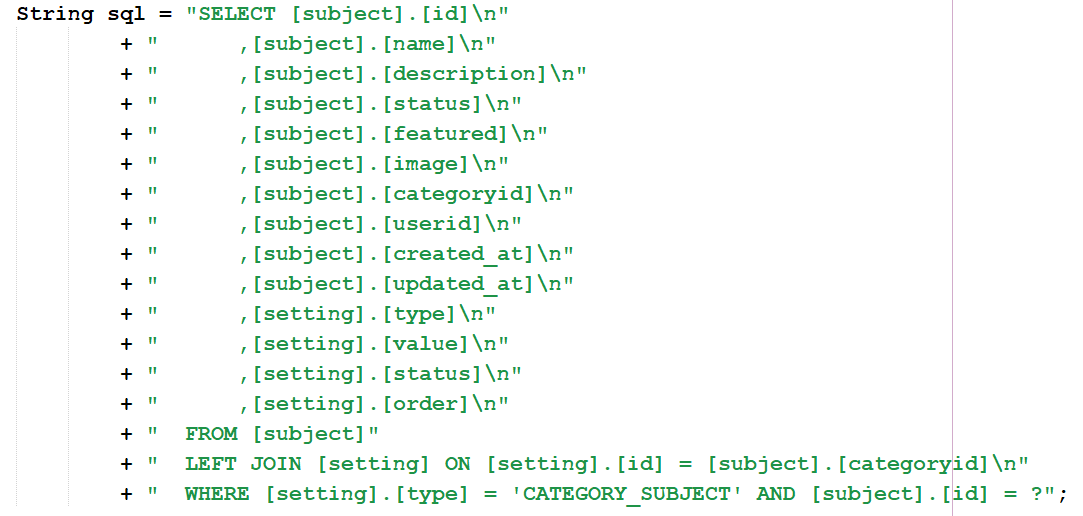
### b. Class Specifications

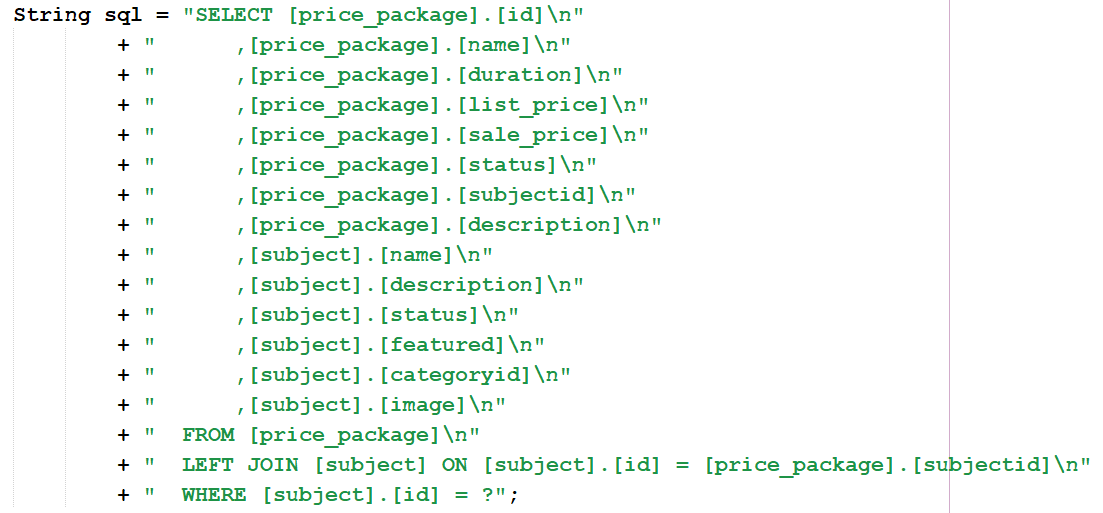
| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **CourseDetail()** | **show subject detail** |
| **2** | **getSubject(int id)** | **get 1 subject by id** |
| **3** | **findBySubjectId(subjectId)** | **get price package by subjectId** |
| **4** | **getFeatureSubjectsByCategory(categoryId)** | **get some subject have feature = true and status = true and same category** |

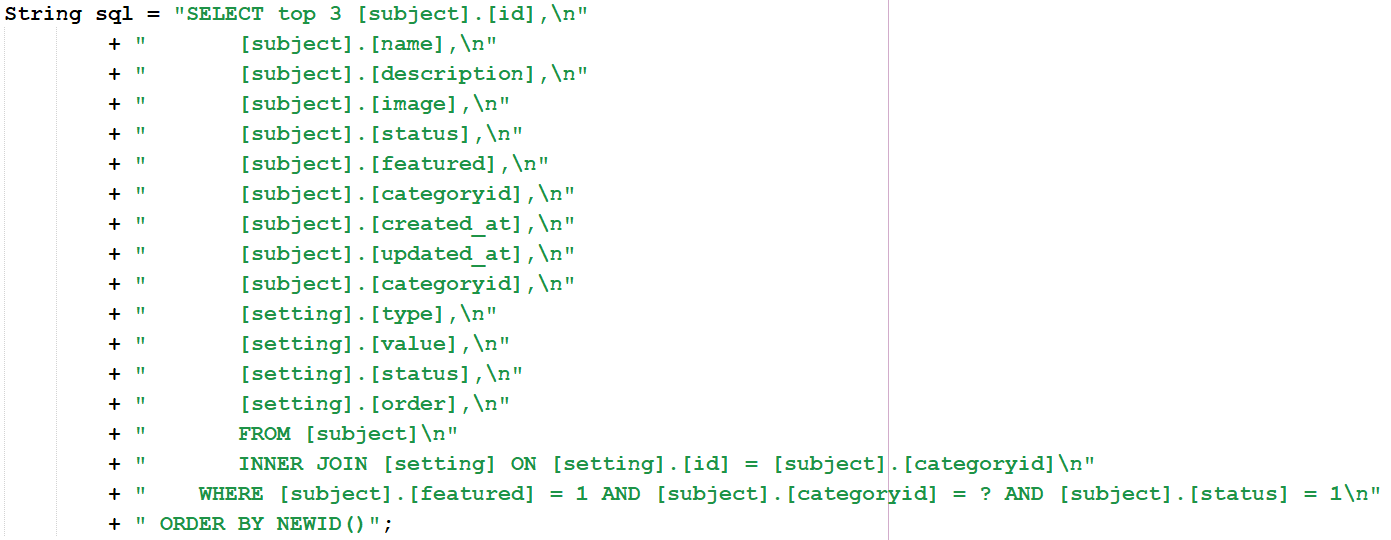
### c. Sequence Diagram(s)



### d. Database queries

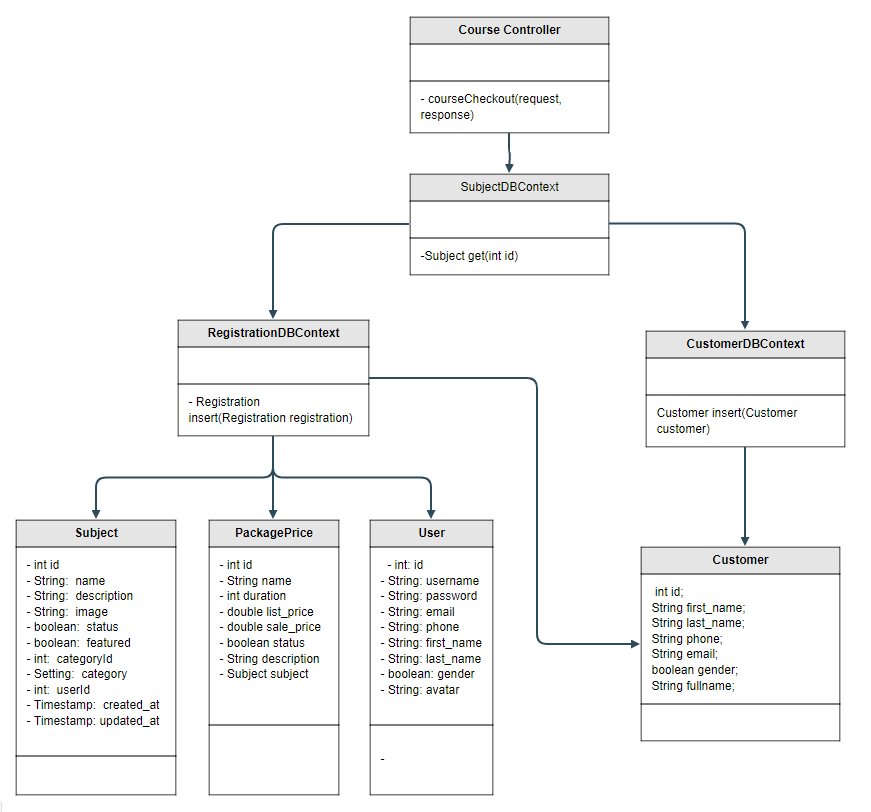






## 12. Public/Course Register

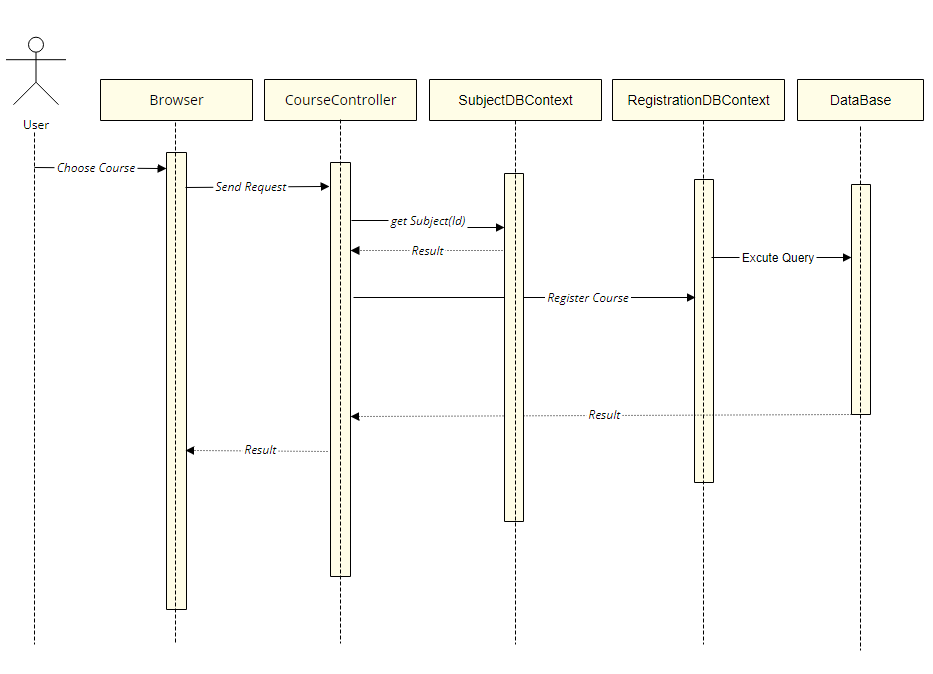
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **Subject get(int id)** | **get subject by id** |
| **2** | **Registration insert(Registration registration)** | **insert information of course registration** |
| **3** | **Customer insert(Customer customer)** | **insert information of new customer** |

### c. Sequence Diagram(s)

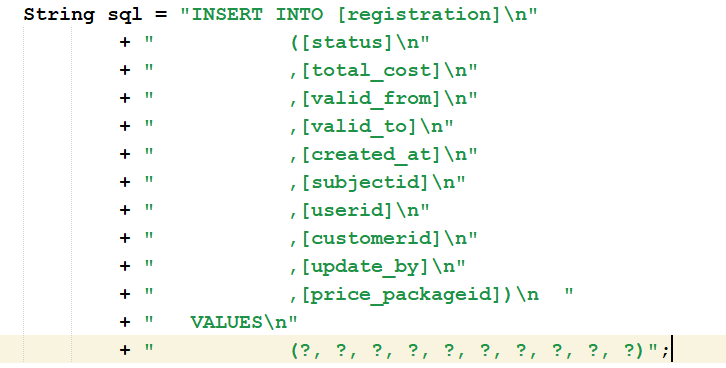


### d. Database queries

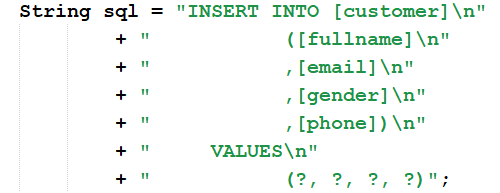
* **Subject get(int id):**



* **Registration insert(Registration registration):**

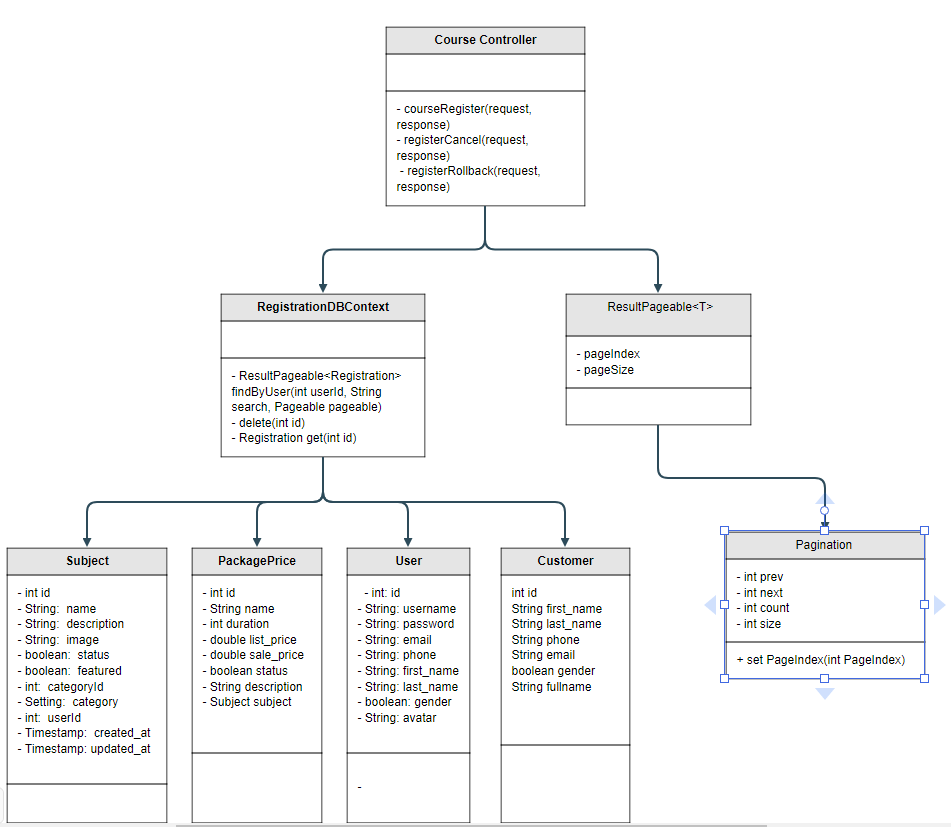
****

* **Customer insert(Customer customer):**

****

## 13. Customer/My Registrations

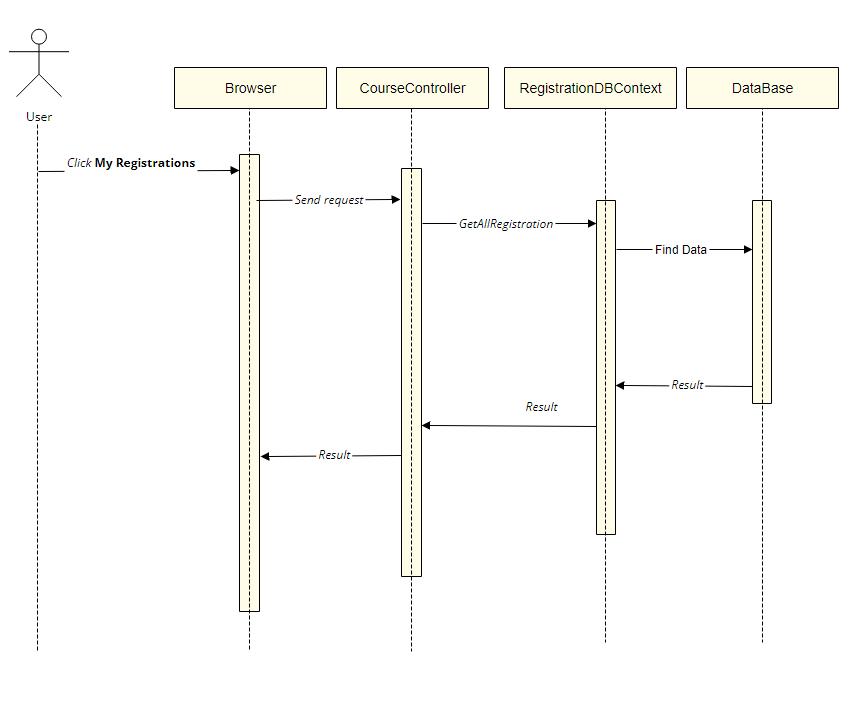
### a. Class Diagram



### b. Class Specifications

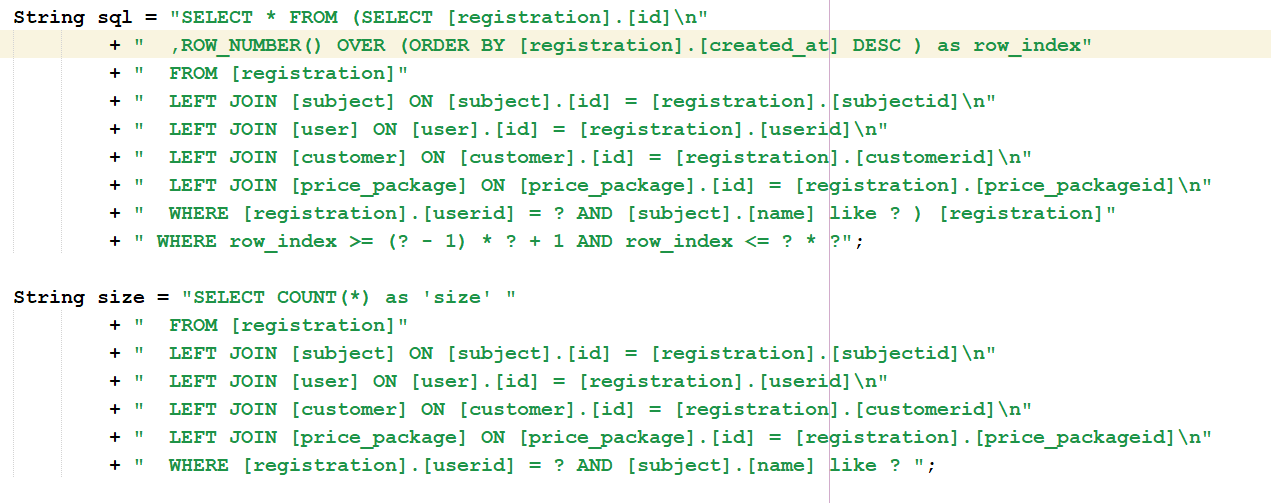
| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **ResultPageable<Registration> findByUser(int userId, String search, Pageable pageable)** | **get all registration by userID with pagination and search registration by subject name** |
| **2** | **delete(int id)** | **cancel registration by registrationId** |
| **3** | **Registration get(int id)** | **get information of registration by registrationId when rollback** |

### c. Sequence Diagram(s)



### d. Database queries

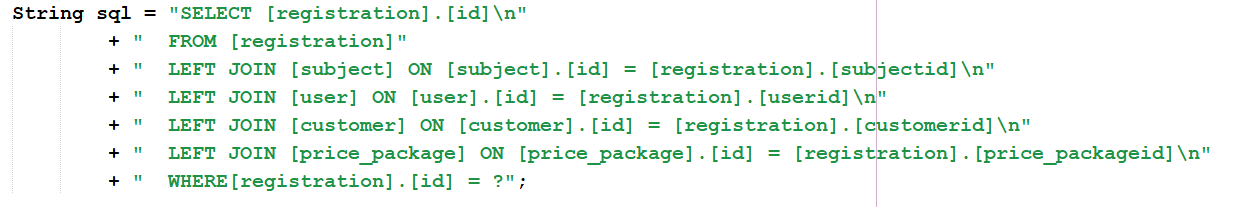
* **ResultPageable<Registration> findByUser(int userId, String search, Pageable pageable):**

****

* **delete(int id):**

****

* **Registration get(int id):**

****

## 14. Customer/My Courses

### a. Class Diagram

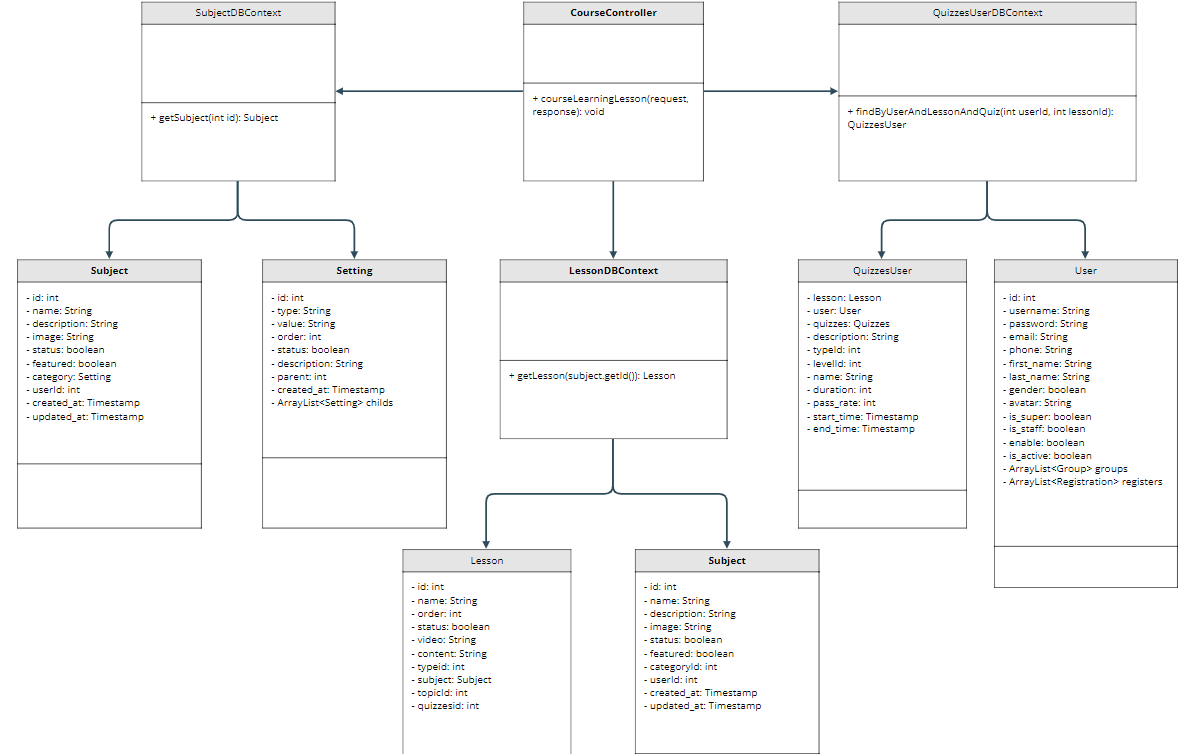
### b. Class Specifications

### c. Sequence Diagram(s)

### d. Database queries

## 15. Customer/Lesson View

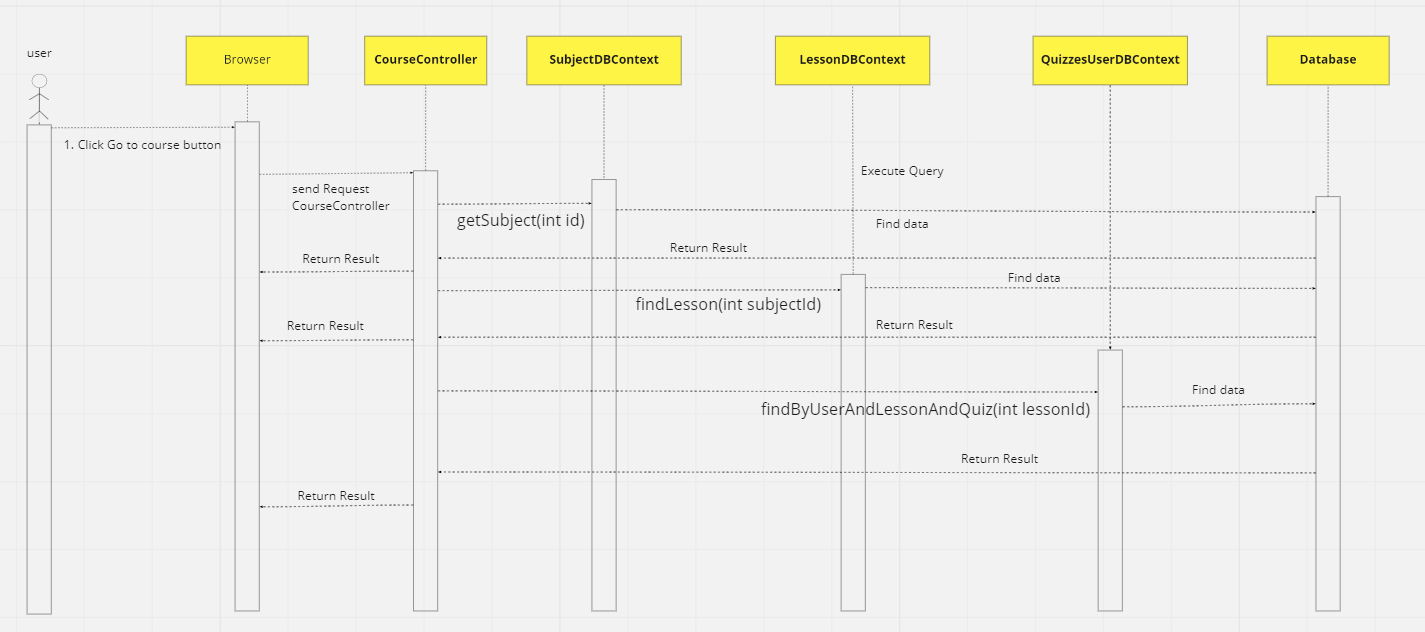
### a. Class Diagram



### b. Class Specifications

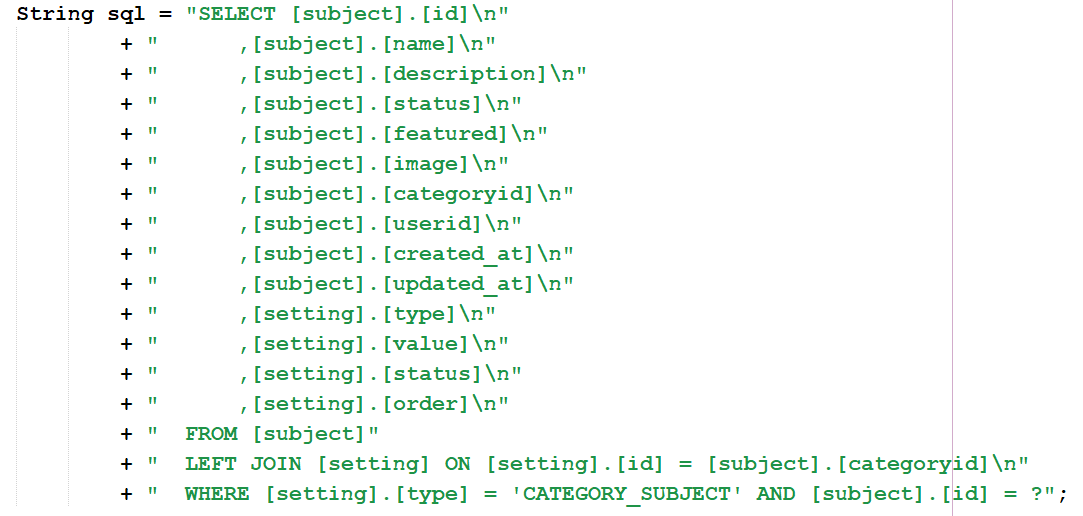
| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getSubject(int id)** | **get 1 subject by id** |
| **2** | **getLesson(int subjectId)** | **get 1 Lesson by subjectId** |
| **3** | **findByUserAndLessonAndQuiz(int userId, int lessonId)** | **get quizz by lessonId** |

### c. Sequence Diagram(s)



### d. Database queries

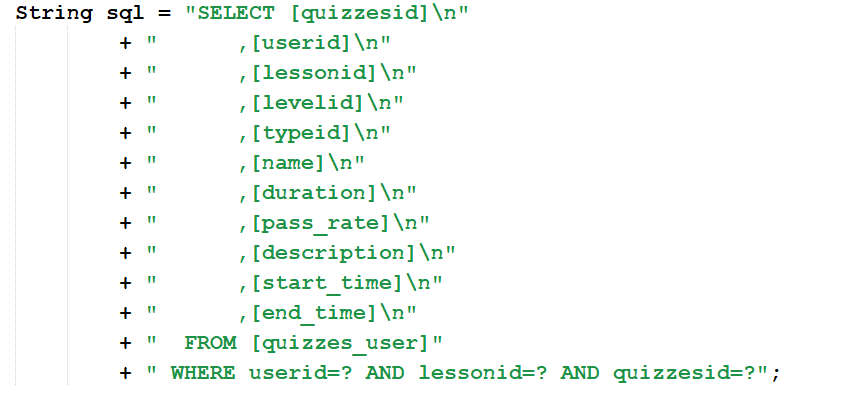
* **getSubject(int id):**



* **getLesson(int subjectId):**

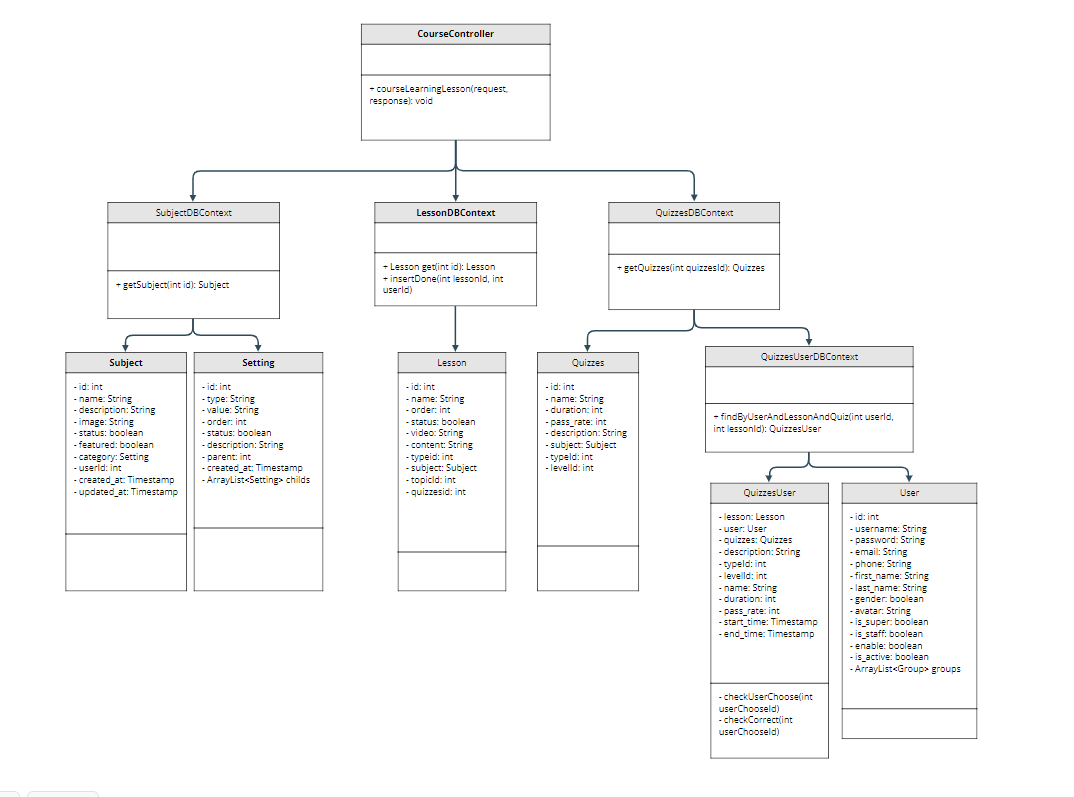
****

* **findByUserAndLessonAndQuiz(int userId, int lessonId):**

****

## 16. Customer/Quiz Lesson

### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getSubject(int id)** | **get subject by id** |
| **2** | **getLesson(int subjectId)** | **get lesson by subjectId** |
| **3** | **findByUserAndLessonAndQuiz(int userId, int lessonId)** | **get quizz by lessonId** |
| **4** | **checkUserChoose(int userChooseId)** | **get answer user choose by userChooseId** |
| **5** | **checkCorrect(int userChooseId)** | **check answer isCorrect by userChooseId** |
| **6** | **insertDone(int lessonId, int userId)** | **get complete lesson by userId** |

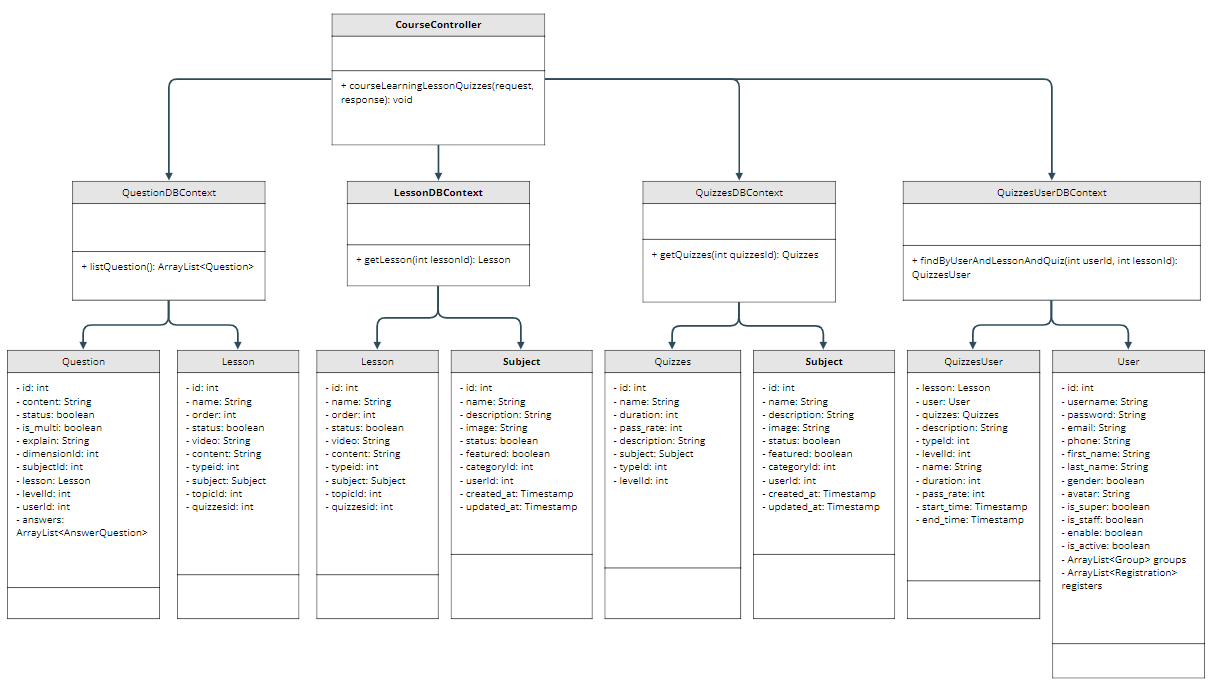
### 

### c. Sequence Diagram(s)

### d. Database queries

## 17. Learning/Quiz Handle

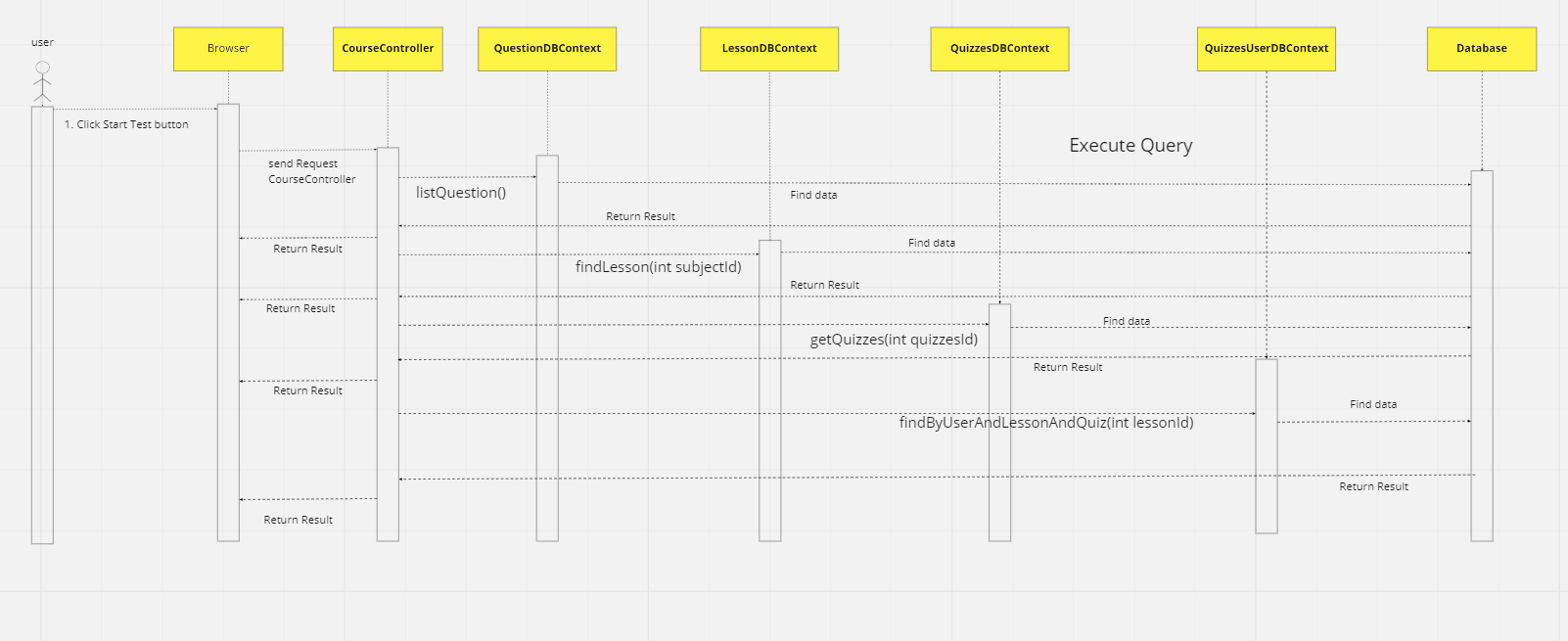
### a. Class Diagram



### b. Class Specifications

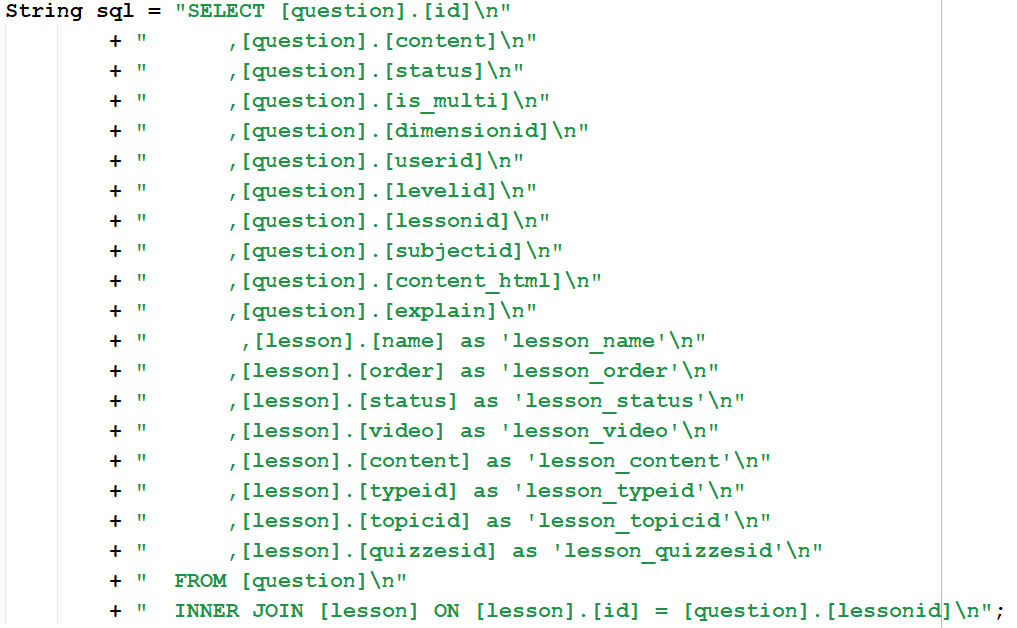
| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **listQuestion()** | **get list question** |
| **2** | **getLesson(int lessonId)** | **get 1 Lesson by id** |
| **3** | **getQuizzes(int quizzesId)** | **get quizzes** |
| **4** | **findByUserAndLessonAndQuiz(int userId, int lessonId)** | **get quizz by lessonId** |

### c. Sequence Diagram(s)



### d. Database queries

* **listQuestion():**

****

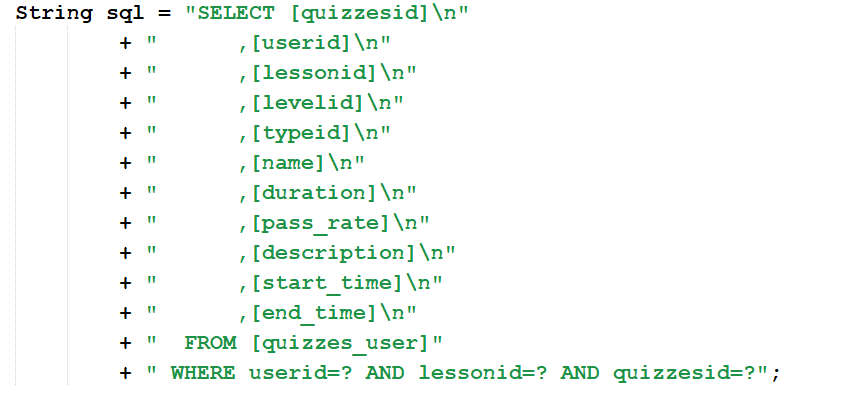
* **getLesson(int lessonId):**

****

* **getQuizzes(int quizzesId):**

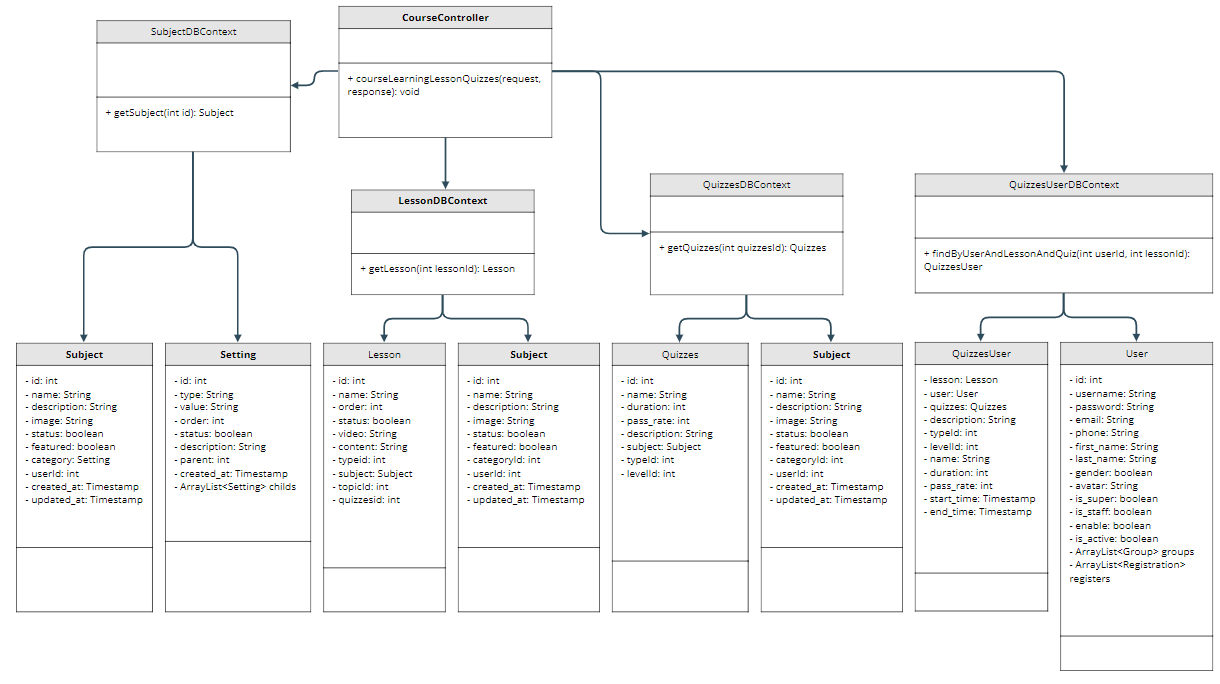
****

* **findByUserAndLessonAndQuiz(int userId, int lessonId):**

****

## 18. Learning/Quiz Review

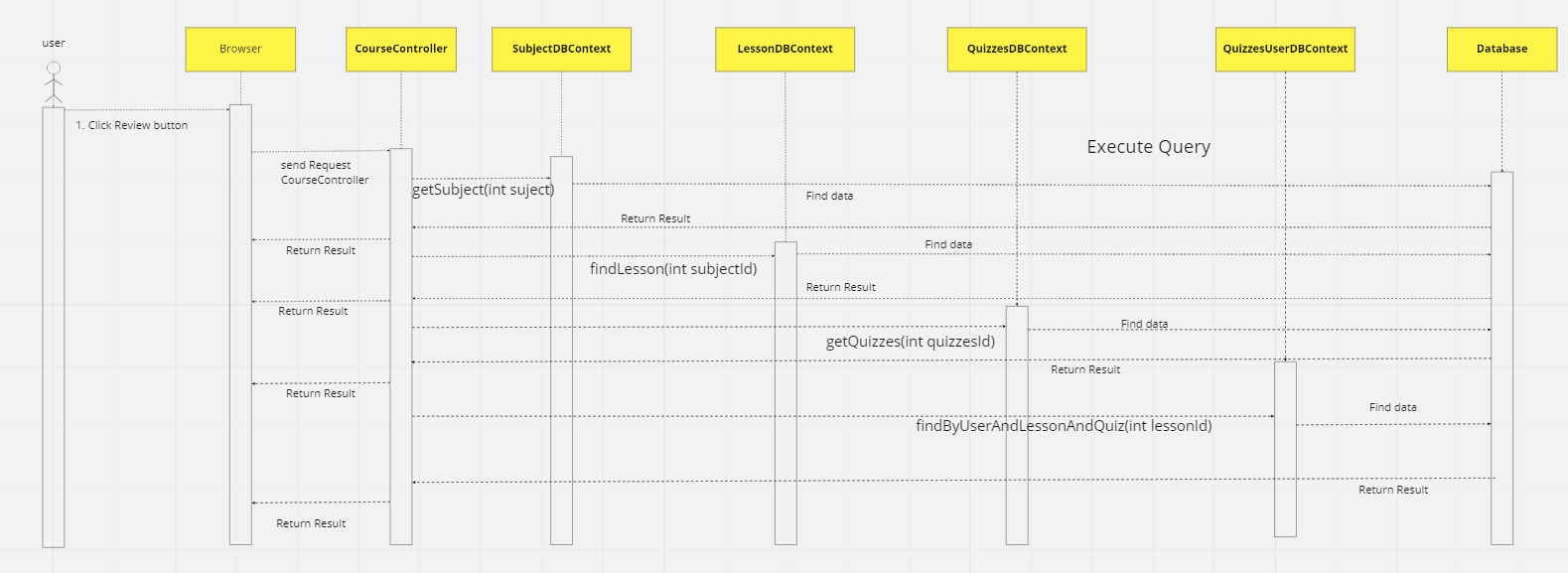
### a. Class Diagram



### b. Class Specifications

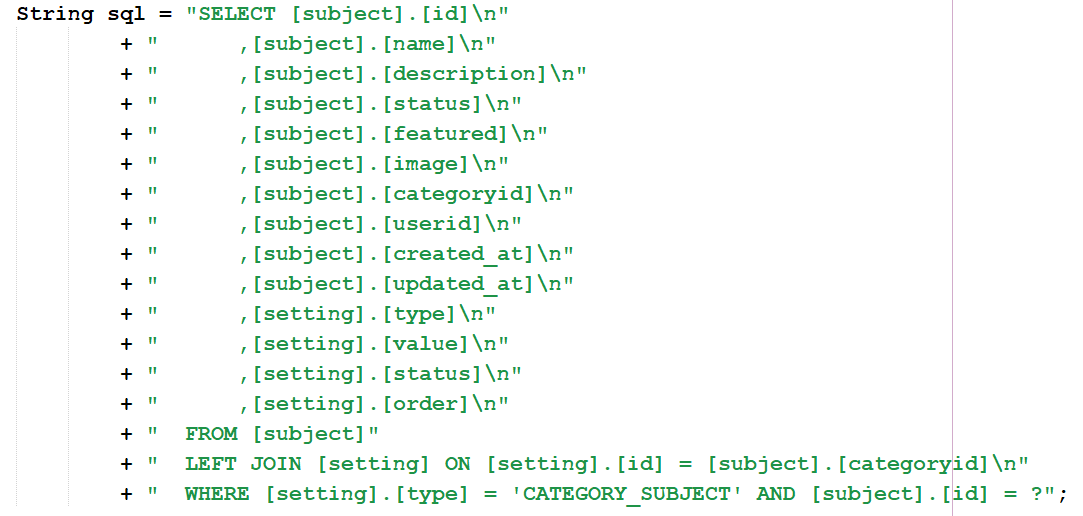
| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getSubject(int id)** | **get 1 subject by id** |
| **2** | **getLesson(int lessonId)** | **get 1 Lesson by id** |
| **3** | **getQuizzes(int quizzesId)** | **get quizzes** |
| **4** | **findByUserAndLessonAndQuiz(int userId, int lessonId)** | **get quizz by lessonId** |

### c. Sequence Diagram(s)



### d. Database queries

* **getSubject(int id):**



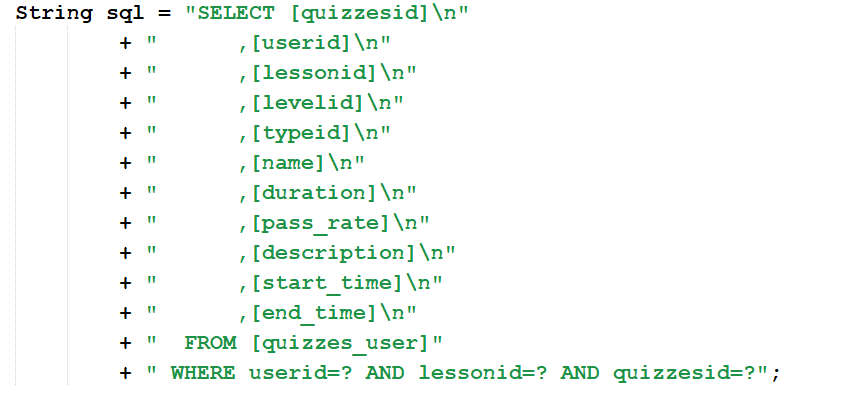
* **getLesson(int lessonId):**

****

* **getQuizzes(int quizzesId):**

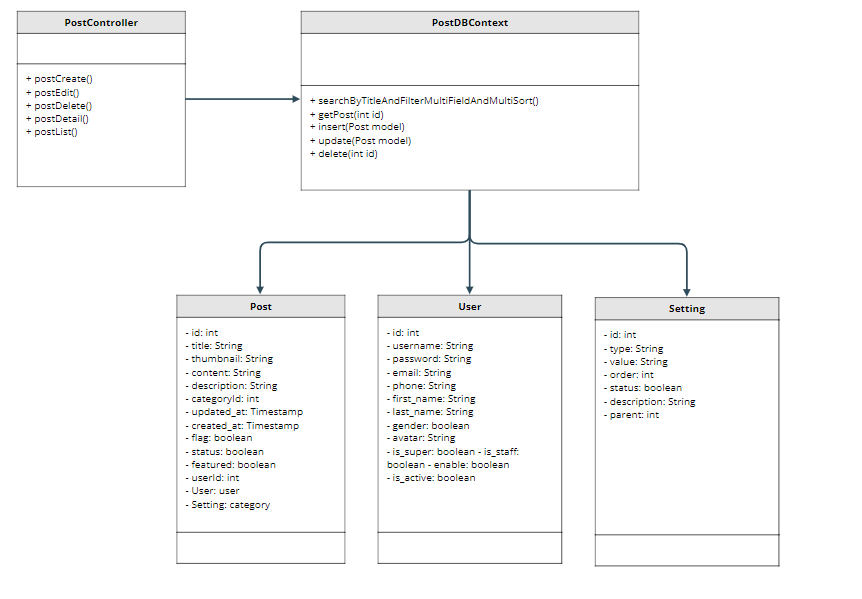
****

* **findByUserAndLessonAndQuiz(int userId, int lessonId):**

****

## 20. Marketing/Post Details

### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **searchByTitleAndFilterMultiFieldAndMultiSort()** | * **get all Post in database** * **User can search Post by title** * **User can filter by Category** * **User can sort by id, title, category, author, featured, status, created, updated** |
| **2** | **getPost(int id)** | **get Post by id** |
| **3** | **insert(Post model)** | **insert information into database** |
| **4** | **update(Postmodel)** | **update information in database** |
| **5** | **delete(int id)** | **delete Post by id** |

### c. Sequence Diagram(s)

### d. Database queries

INSERT INTO [post]

([title], [thumbnail], [content], [featured], [flag], [status], [created\_at],

[updated\_at], [userid], [categoryId], [description])

VALUES (?,?,?,?,?,?,?,?,?,?,?)

UPDATE [post]

SET [title] = ?, [thumbnail] = ?, [content] = ?, [featured] = ?, [flag] = ?, [status] = ?,

[created\_at] = ?, [updated\_at] = ? , [userid] = ?, [categoryId] = ?, [description] = ?

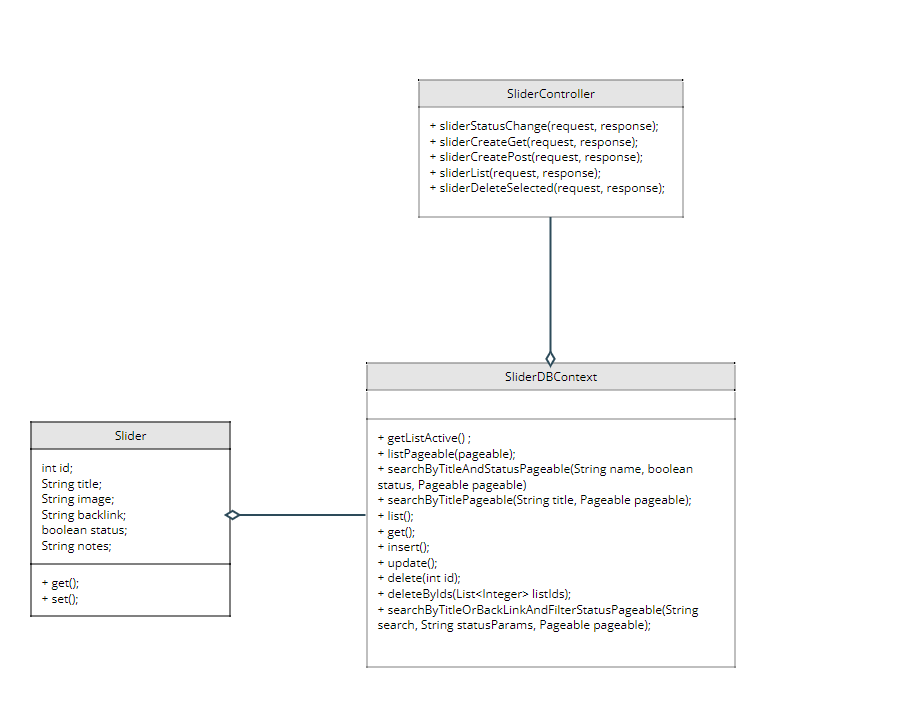
WHERE [id] = ?

DELETE FROM [post]

WHERE [post].[id]

## 21. Marketing/Sliders List

### a. Class Diagram



### b. Class Specifications

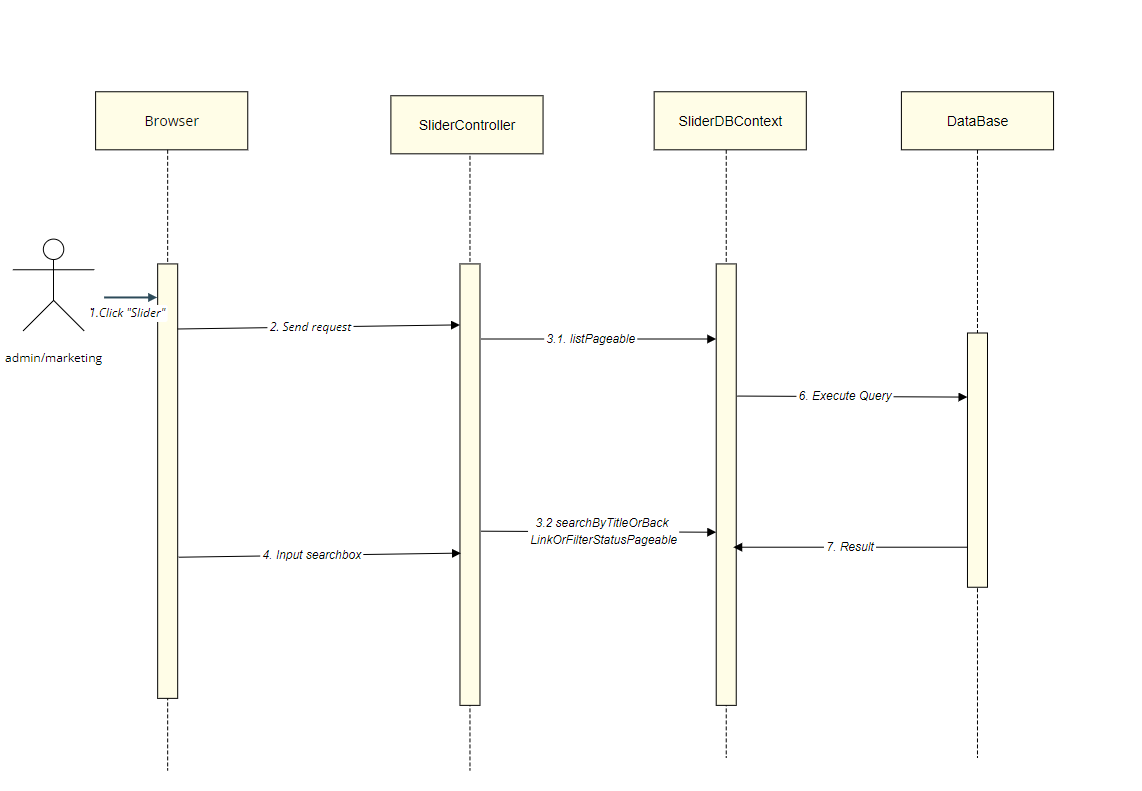
**SliderController:**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **sliderStatusChange** | **request get status slider** |
| **2** | **sliderCreateGet** | **request get slider** |
| **3** | **sliderCreatePost** | **post request slider** |
| **4** | **sliderList** | **request list slider** |
| **5** | **sliderDelete** | **request delete slider** |

**SliderDBContext:**

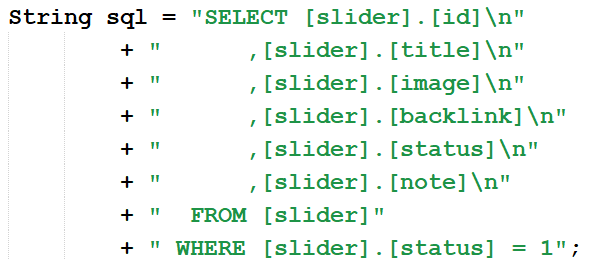
| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getListActive** | **get all list slider have status active()** |
| **2** | **listPageble** | **get all list slider pagination** |
| **3** | **searchByTitleAndStatusPageable** | **get list by search title or status pagination** |
| **4** | **searchByTitlePageable** | **get list by search title pagination** |
| **5** | **list** | **get all list slider** |
| **6** | **insert** | **insert slider** |
| **7** | **update** | **update slider** |
| **8** | **delete** | **delete slider** |
| **9** | **deleteByIds** | **delete multiple id** |
| **10** | **searchByTitleOrBackLinkOrFilterStatusPageable** | **get list by search title or status or backlink pagination** |

### c. Sequence Diagram(s)

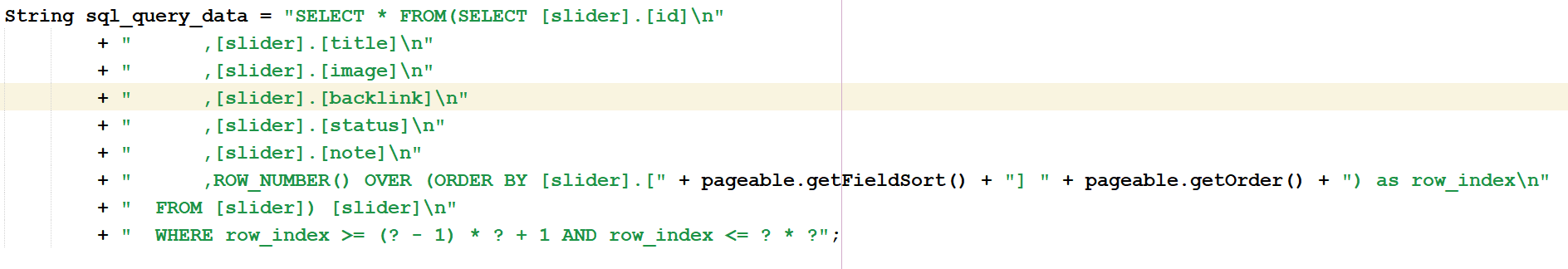


### d. Database queries

#### Get slider active from database:

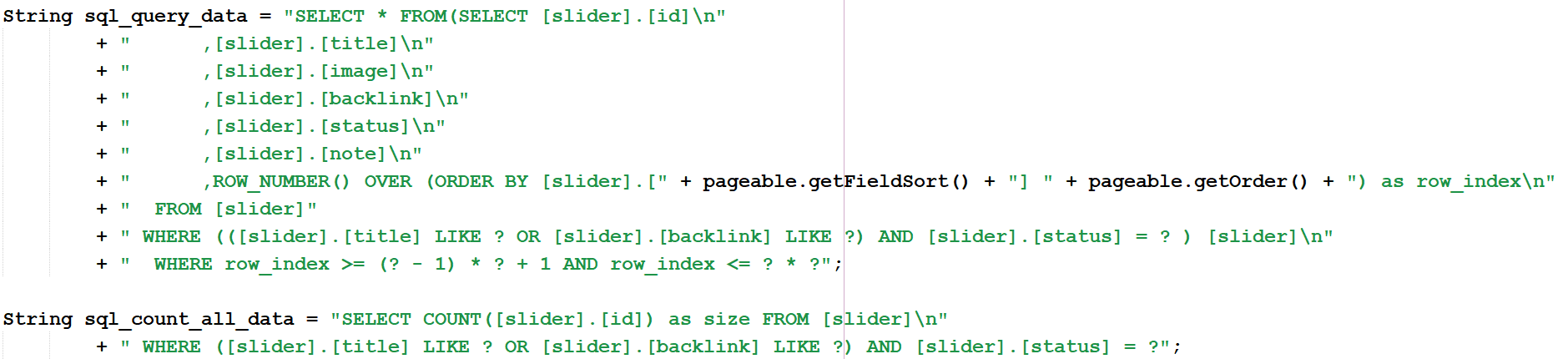


#### Get list slider pageable from database:

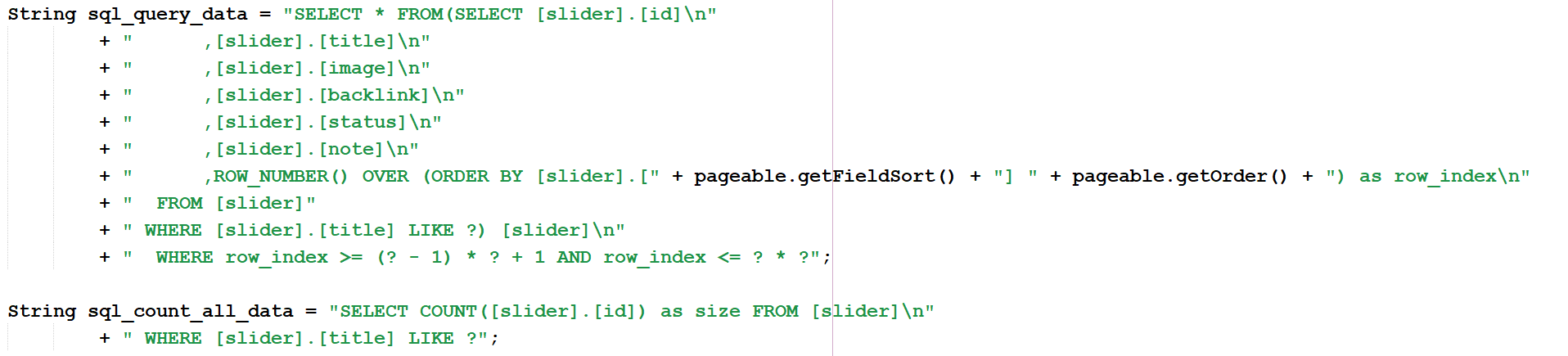




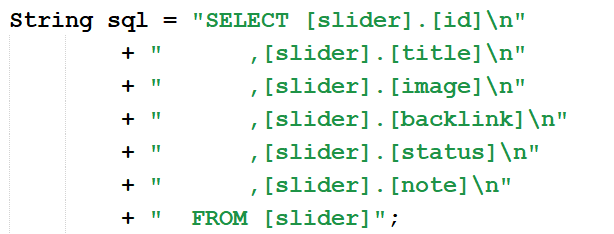
#### Get list slider by search Title And Status Pageable from database:



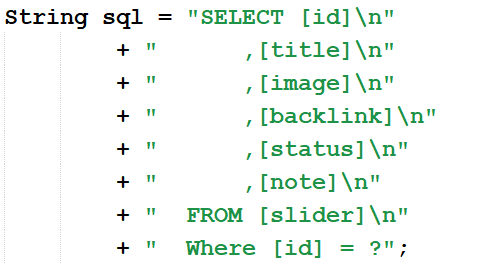
#### Get list slider by search Title Pageable from database:



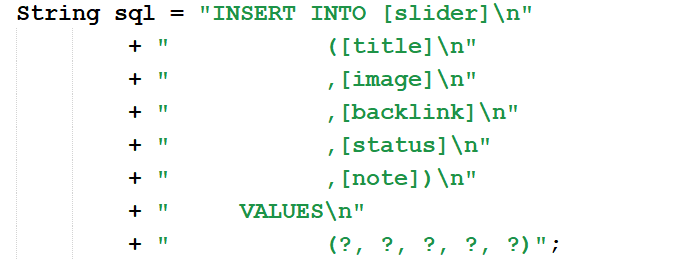
#### Get list all slider from database:

****

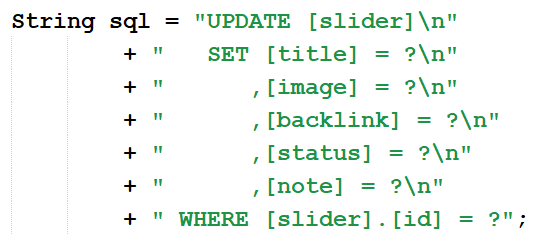
#### Get slider by id from database:



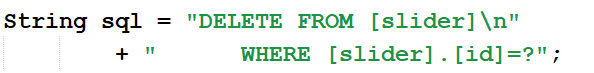
#### Insert slider from database:



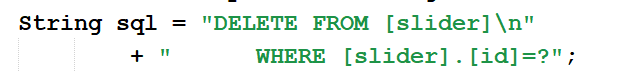
#### Update slider from database:



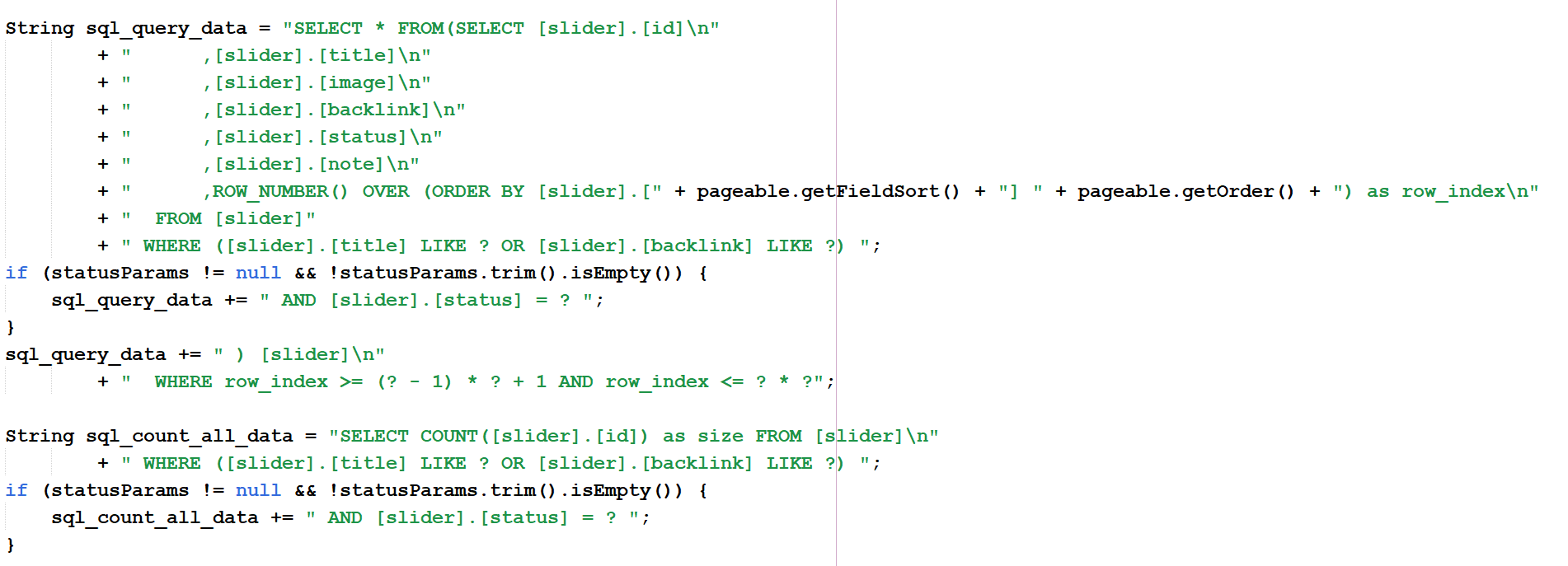
#### Delete slider from database:



#### Delete by Ids slider from database:

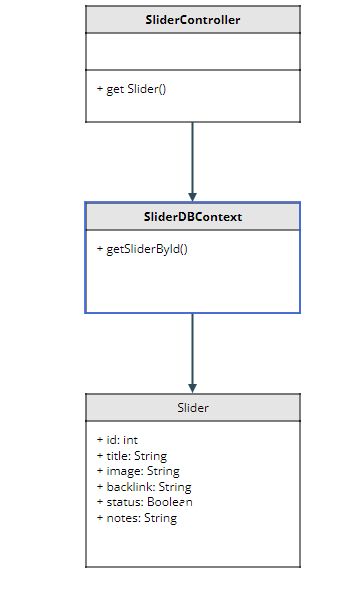


#### Get list slider by search Title Or Status Or Backlink Pageable from database:



## 22. Marketing/Slider Details

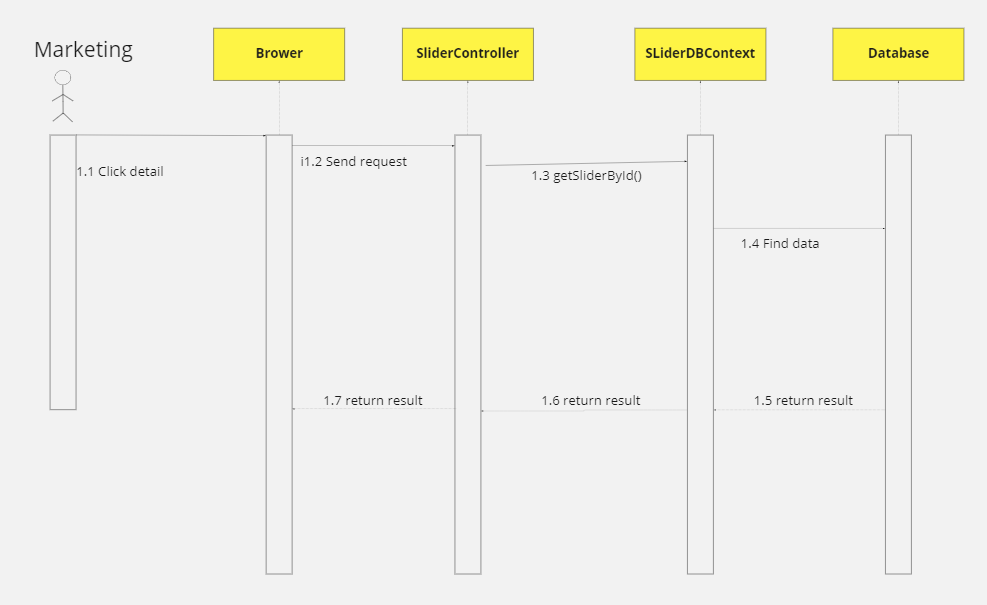
### a. Class Diagram



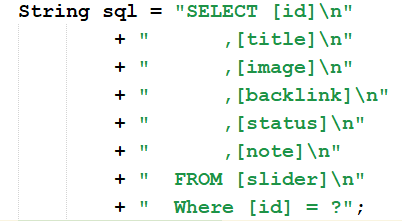
### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getSliderById()** | **get slider by id** |

### c. Sequence Diagram(s)

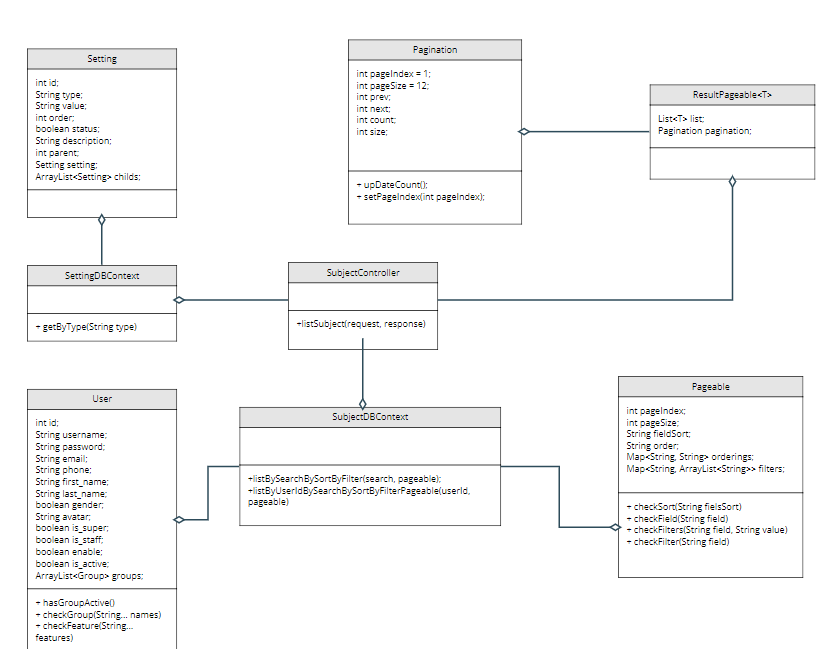


### d. Database queries

****

## 23. Course Content/Subjects List

### a. Class Diagram

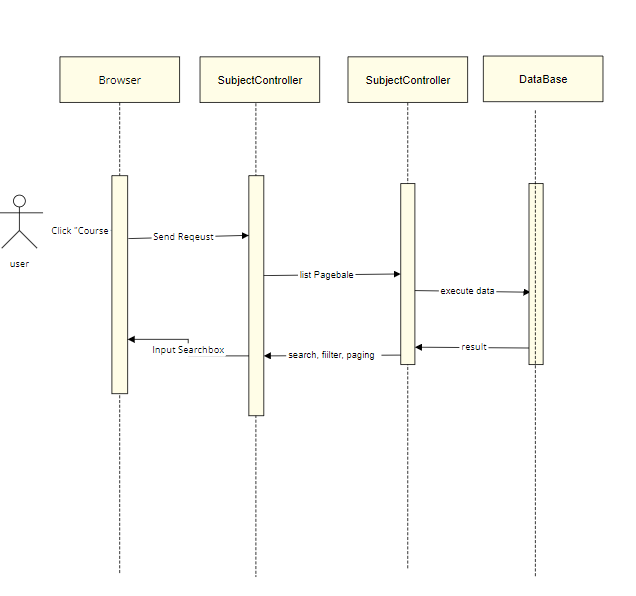


### b. Class Specifications

**SubjectDBContext**

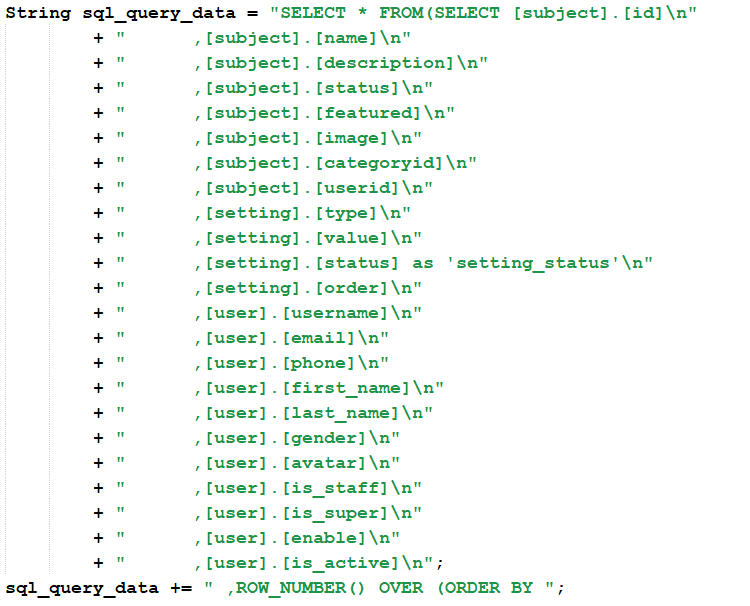
| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **listBySearchBySortByFilter** | **get list subject by search by sort by filter and pagination** |
| **2** | **listByUserIdBySearchBySortByFilterPageable** | **get list subject by search by sort by filter and pagination of user expert** |

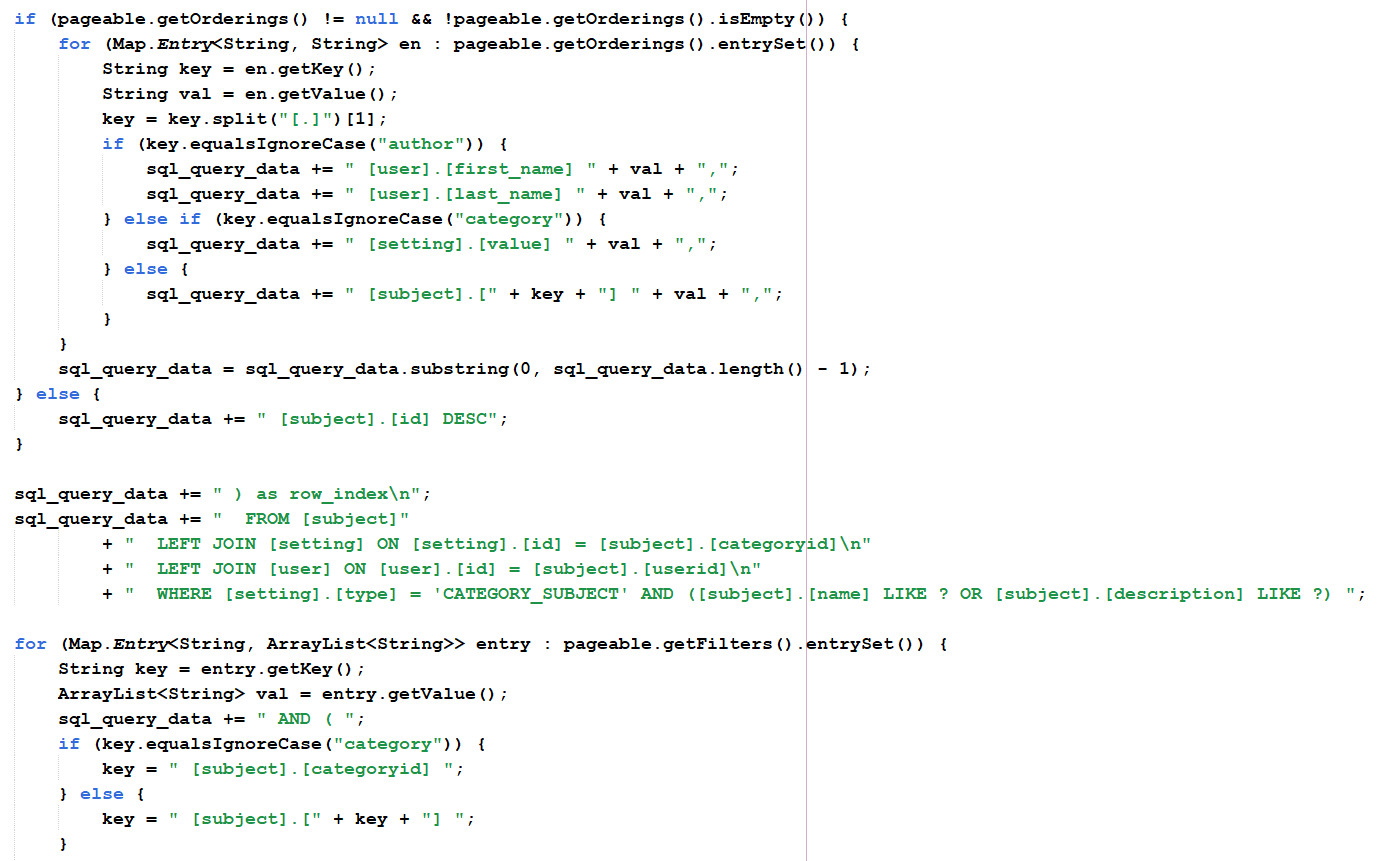
### c. Sequence Diagram(s)

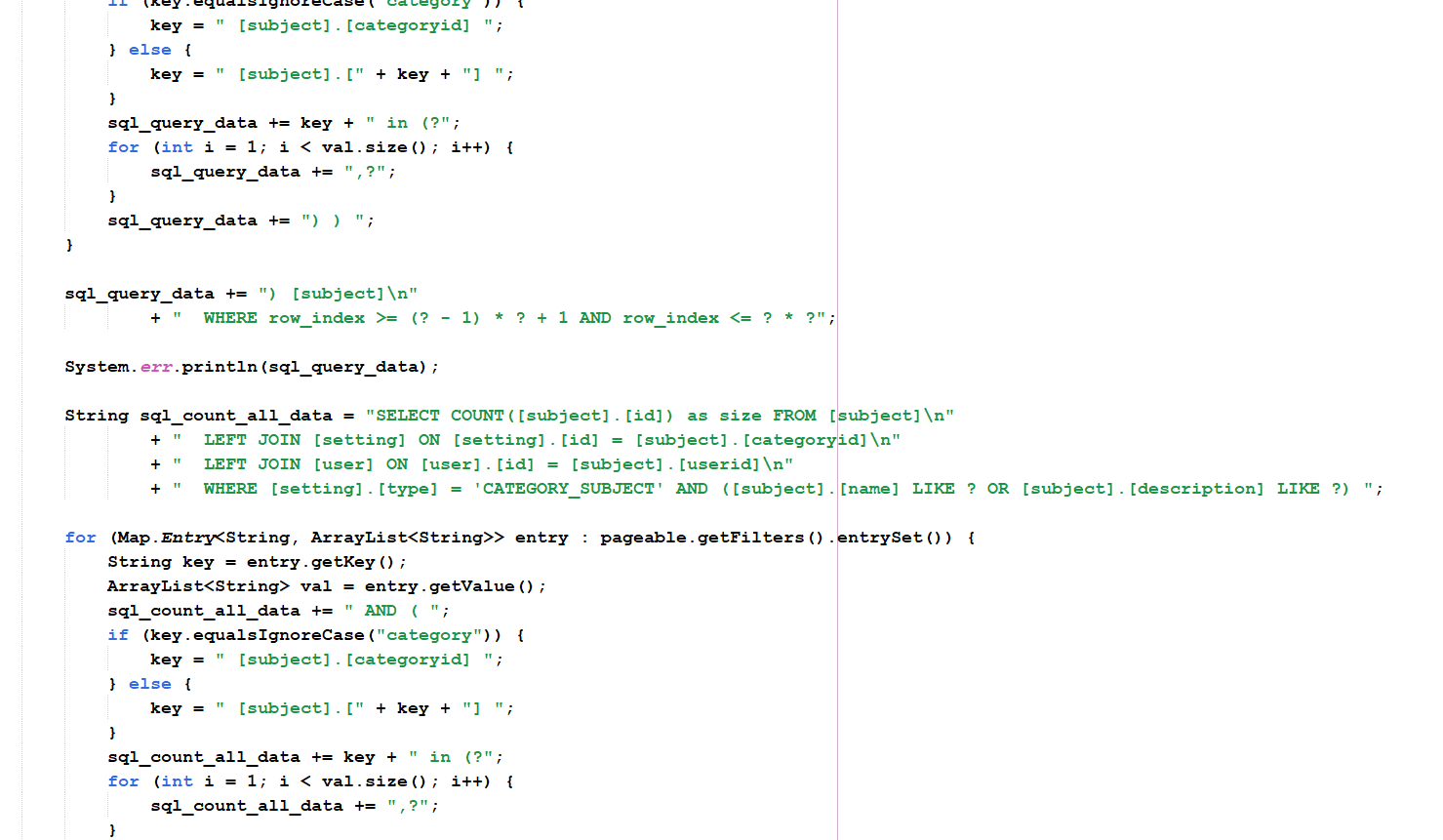


### d. Database queries

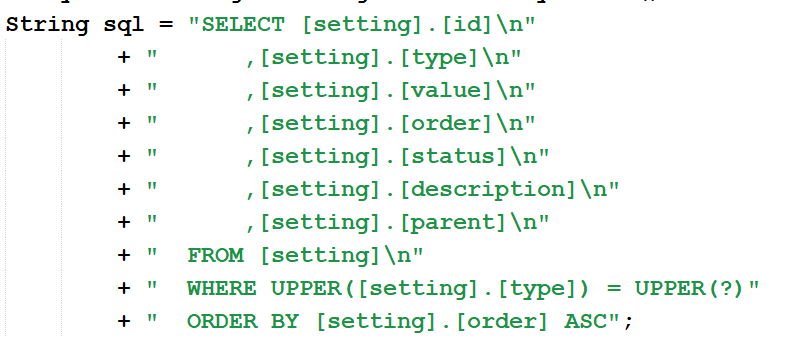
#### Get list subject by search by sort by filter and pagination





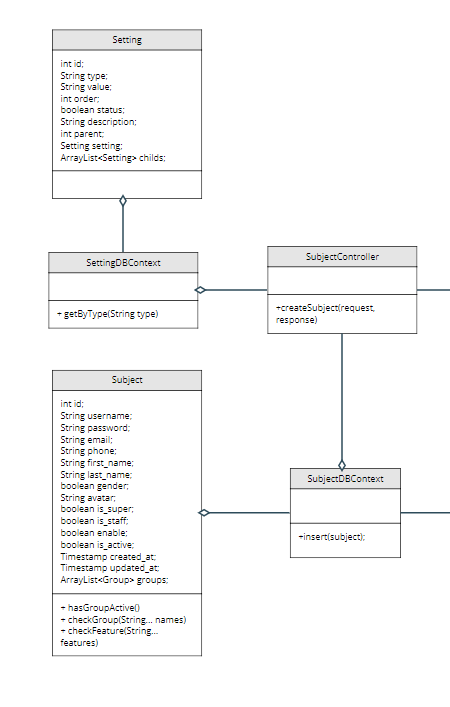


#### Get list setting by type



## 24. Course Content/New Subject

### a. Class Diagram



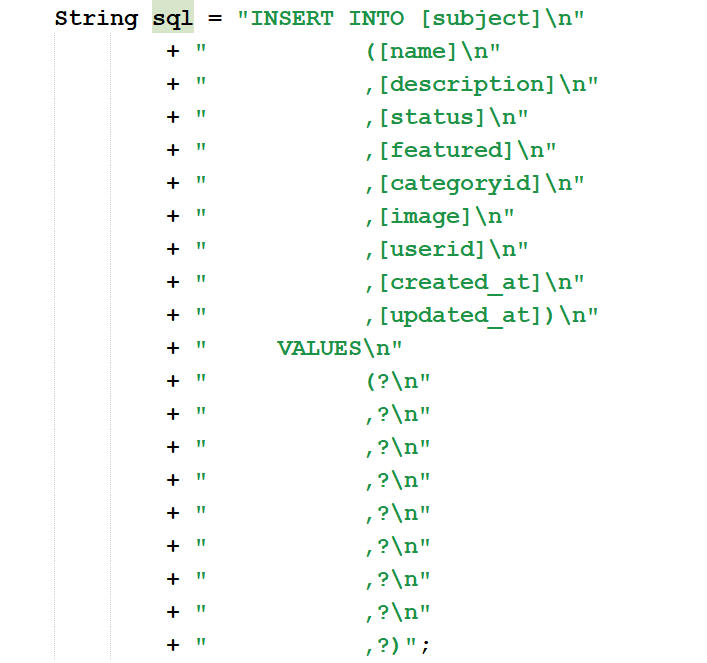
### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **inset** | **insert Subject to database** |

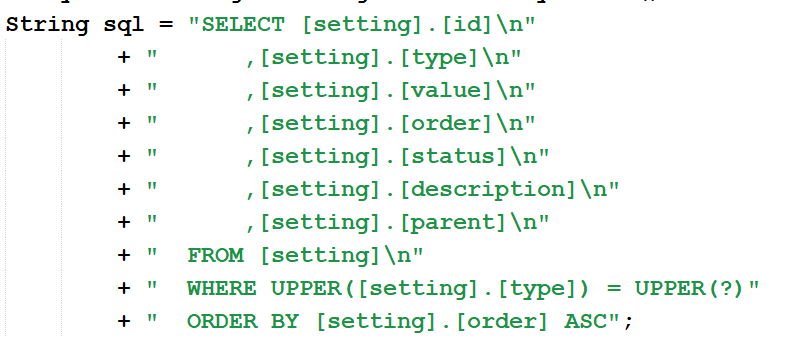
### c. Sequence Diagram(s)

### d. Database queries

#### Insert Subject to database



#### Get list setting by type



## 25. Course Content/Subject Details

### a. Class Diagram

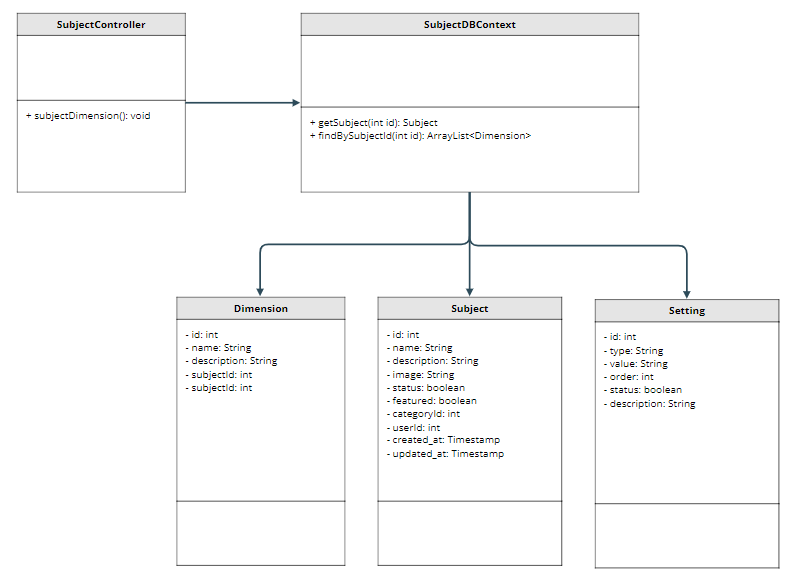
### b. Class Specifications

### c. Sequence Diagram(s)

### d. Database queries

## 26. Course Content/Subject Dimension

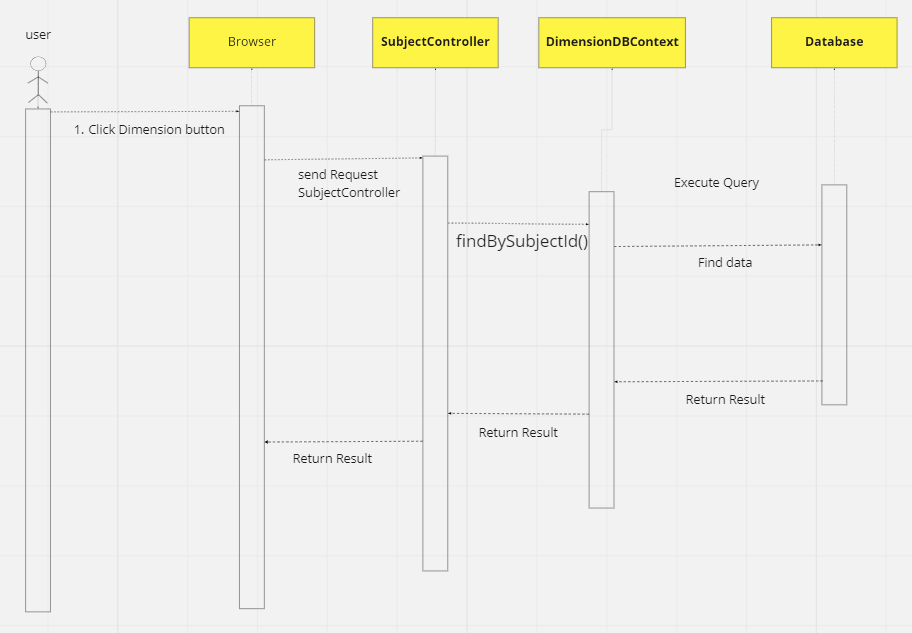
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getSubject(int id)** | **get 1 subject by id** |
| **2** | **findBySubjectId(subject.getId())** | **get Dimension by subjectId** |

### c. Sequence Diagram(s)



### d. Database queries

* **SELECT \***

**FROM [subject] LEFT JOIN [setting] ON [setting].[id] = [subject].[categoryid]**

**LEFT JOIN [user] ON [user].[id] = [subject].[userid]**

**WHERE [setting].[type] = 'CATEGORY\_SUBJECT' AND [subject].[id] = ?**

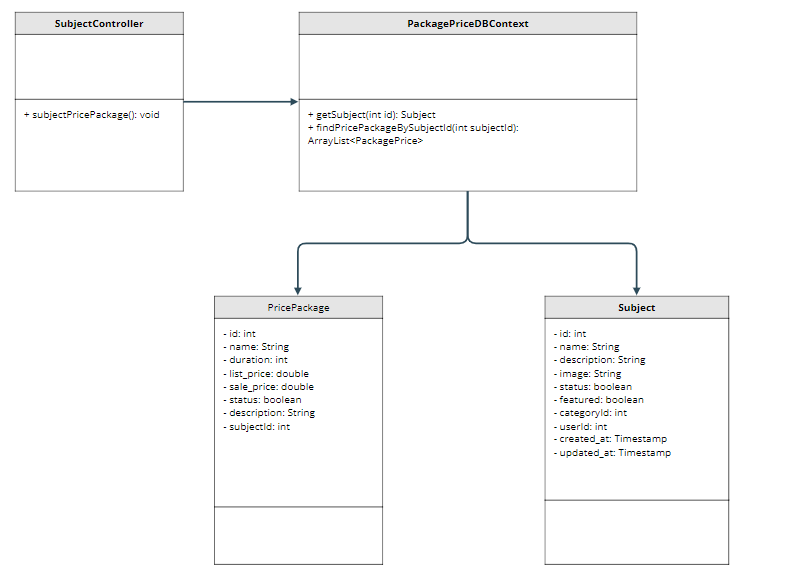
* **SELECT \***

**From [dimension] INNER JOIN [subject] ON [subject].[id] = [dimension].[subjectid]**

**WHERE [dimension].[subjectid] = ?**

## 27. Course Content/Price Package

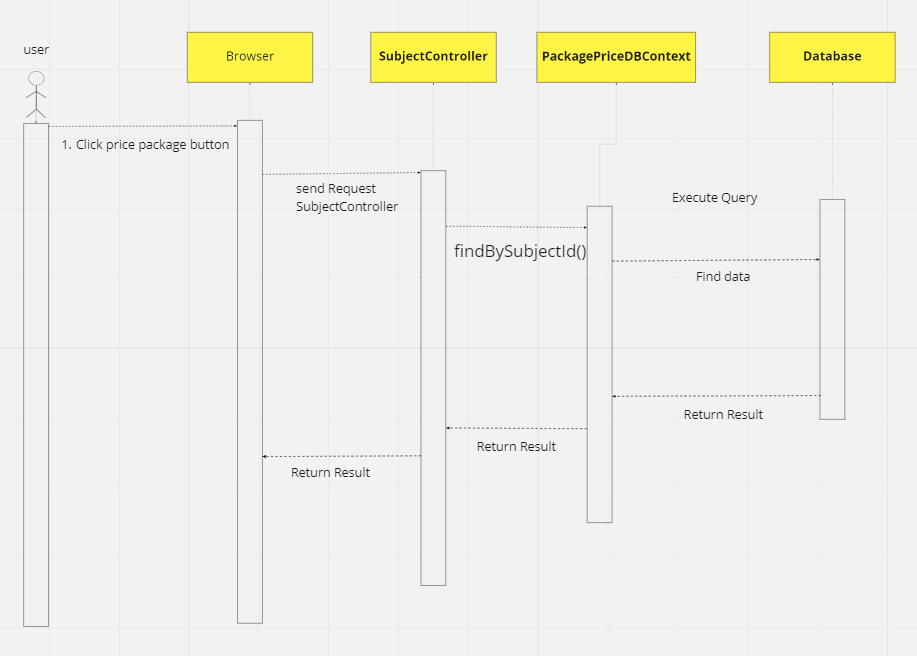
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getSubject(int id)** | **get 1 subject by id** |
| **2** | **findPricePackageBySubjectId(subject.getId())** | **get list Price Package by subjectId** |

### c. Sequence Diagram(s)



### d. Database queries

* **SELECT \***

**FROM [subject] LEFT JOIN [setting] ON [setting].[id] = [subject].[categoryid]**

**WHERE [setting].[type] = 'CATEGORY\_SUBJECT' AND [subject].[id] = ?**

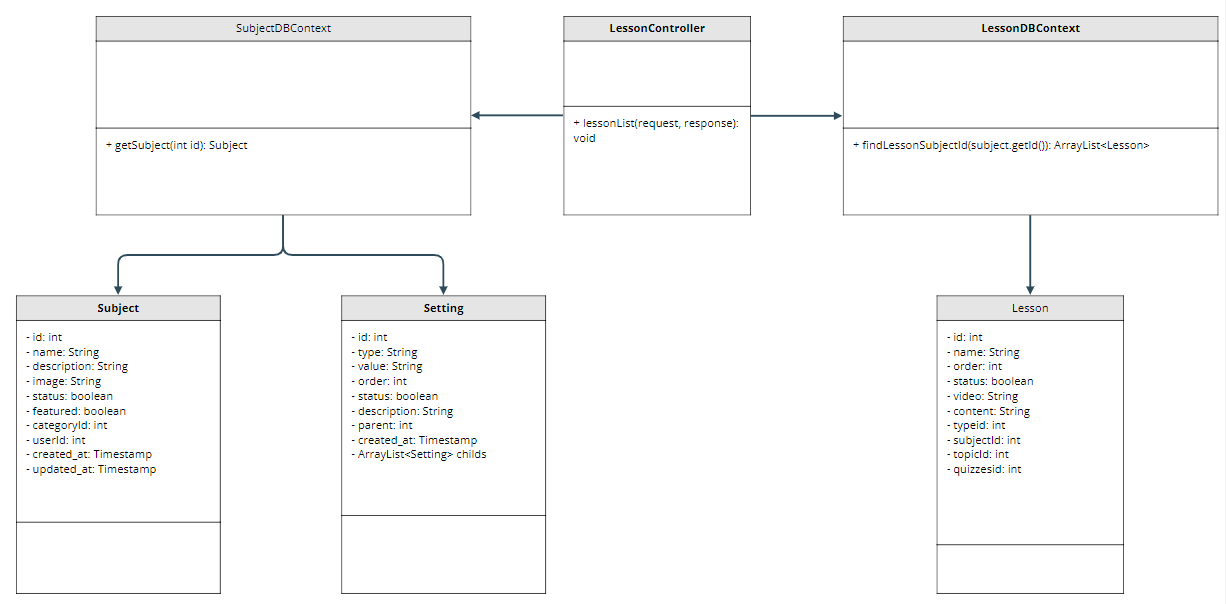
* **SELECT \***

**FROM [price\_package] LEFT JOIN [subject] ON [subject].[id] = [price\_package].[subjectid]**

**WHERE [subject].[id] = ?**

## 28. Course Content/Subject Lessons

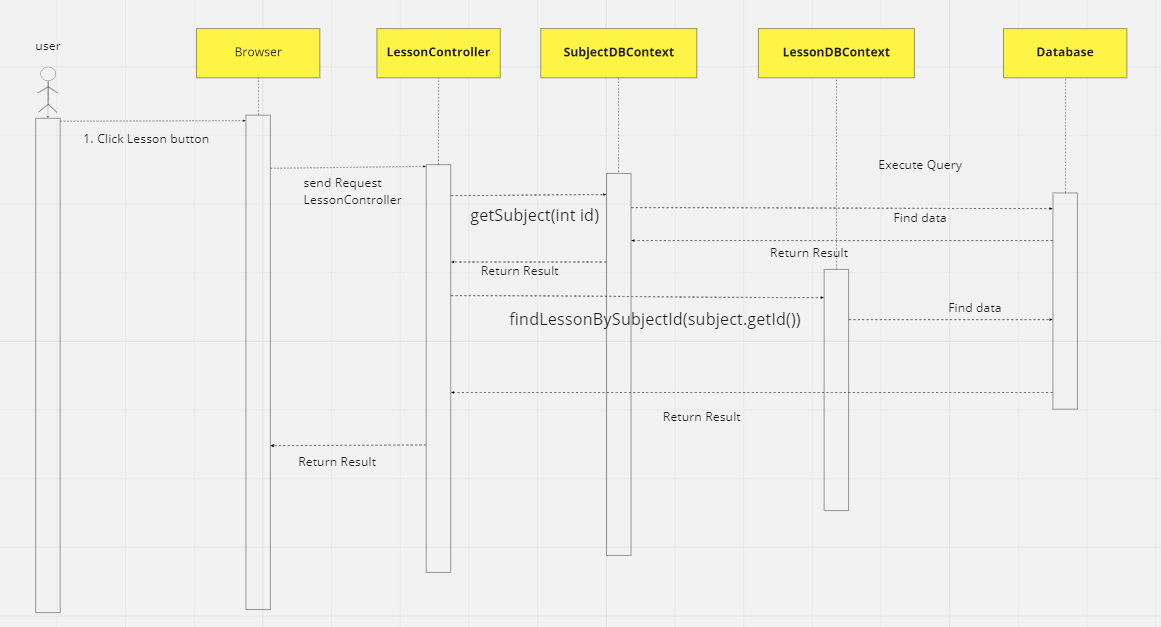
### a. Class Diagram



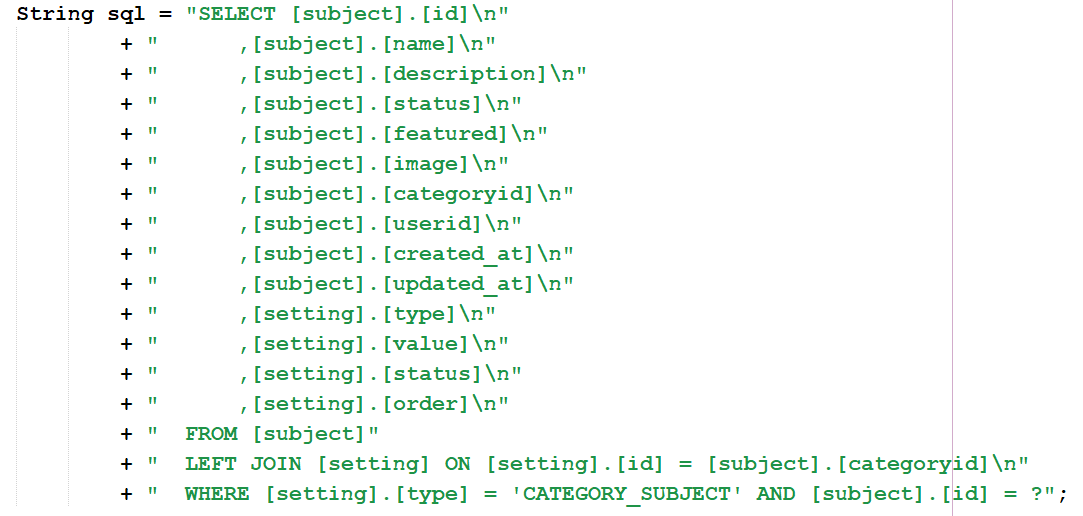
### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getSubject(int id)** | **get 1 subject by id** |
| **2** | **findLessonBySubjectId(subject.getId())** | **get list Lesson by subjectId** |

### c. Sequence Diagram(s)



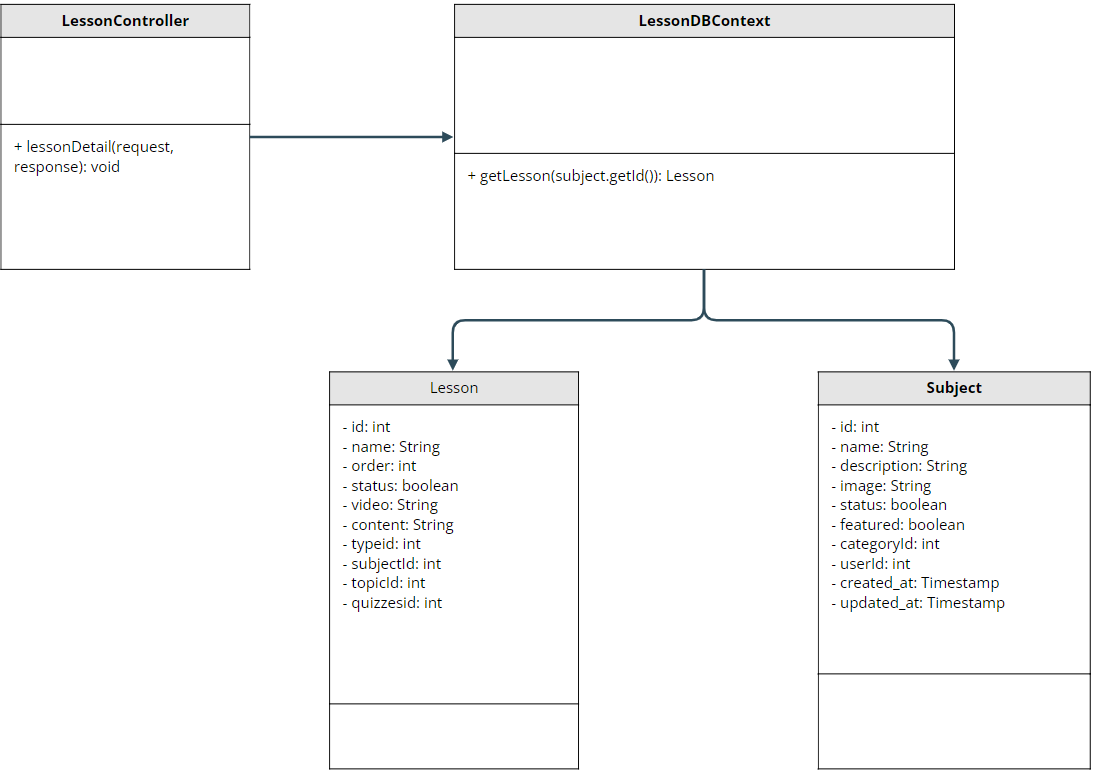
### d. Database queries





## 29. Course Content/Lesson Details

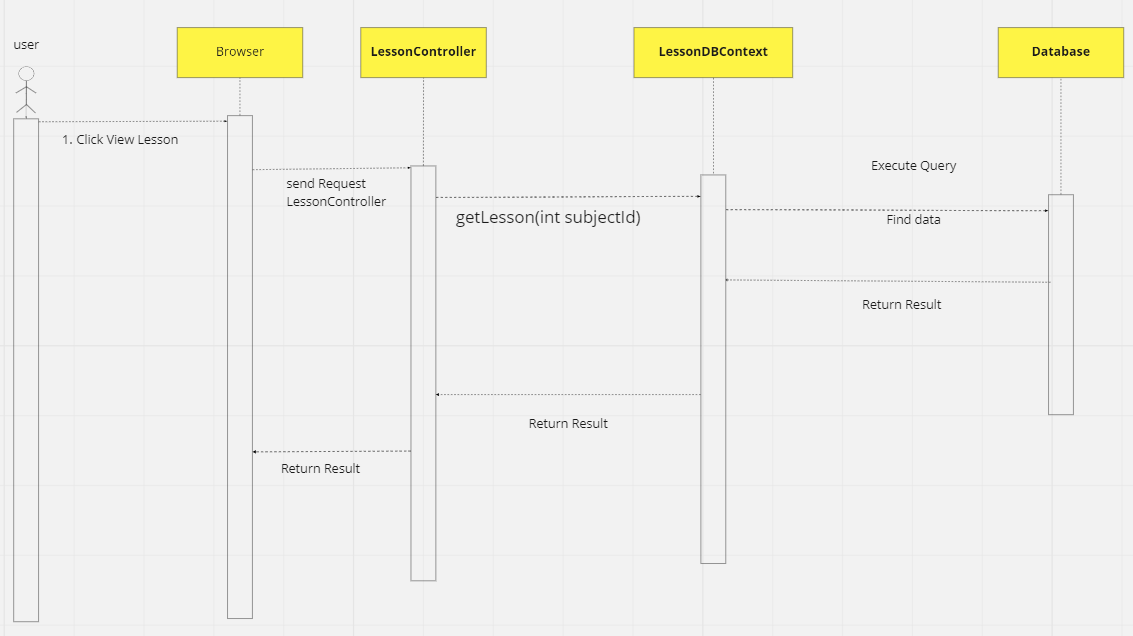
### a. Class Diagram



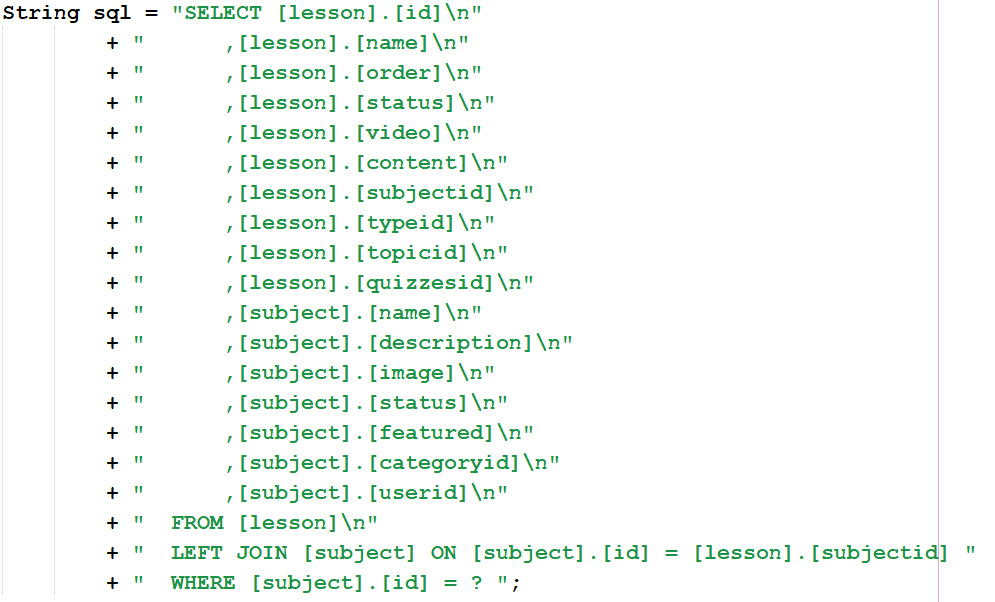
### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getLesson(int subjectId)** | **get 1 lesson by subjectId** |

### c. Sequence Diagram(s)

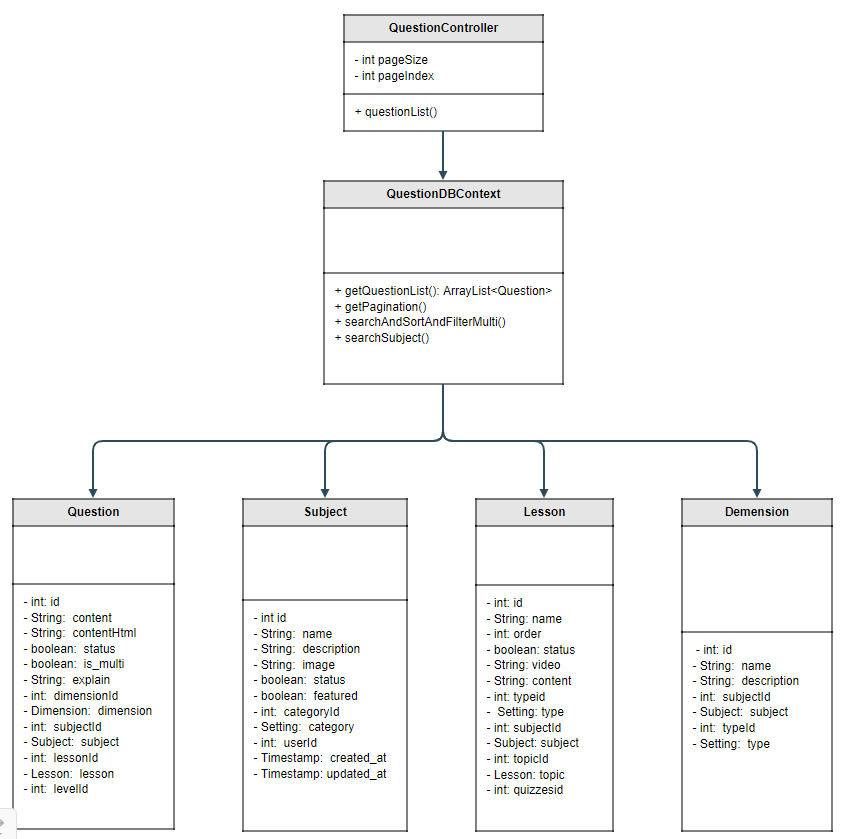


### d. Database queries



## 30. Test Content/Questions List

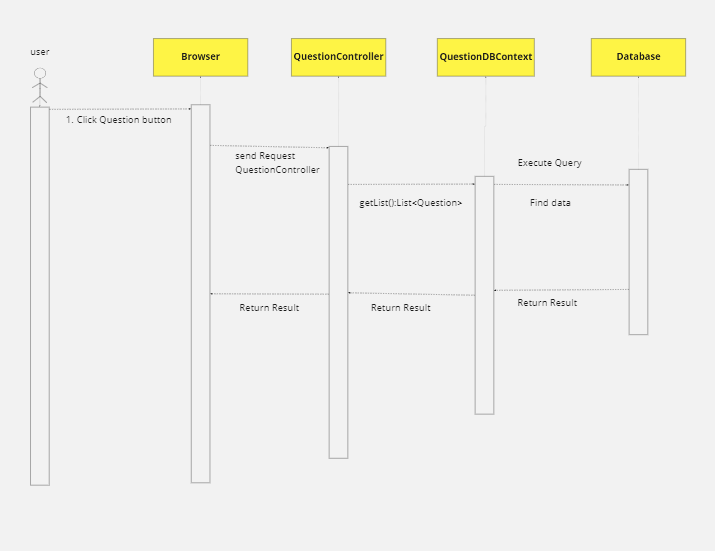
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getQuestionList** | **get all list question have status active()** |
| **2** | **getPagination** | **get all list question pagination** |
| **3** | **searchByTitleAndFiltetMultiFieldAndMultiSort()** | * **User can search Question by content** * **User can sort Question by properties** * **User can filter by Category** |

### c. Sequence Diagram(s)



### d. Database queries

* **getQuestionList:**

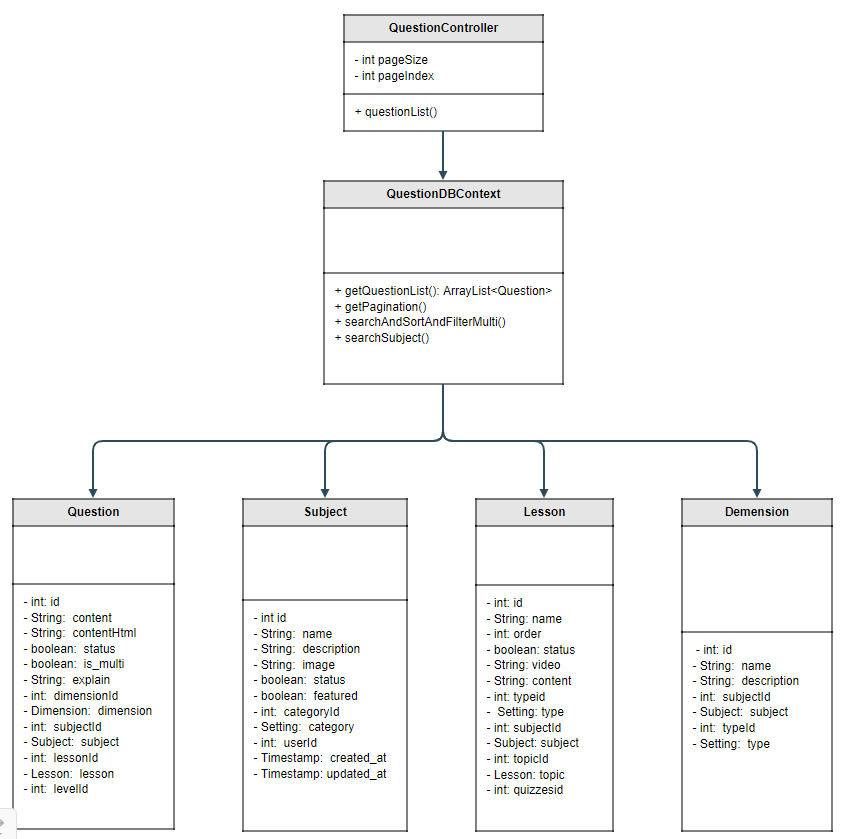
****

* **searchByTitleAndFiltetMultiFieldAndMultiSort():**

****

## 31. Test Content/Questions Details

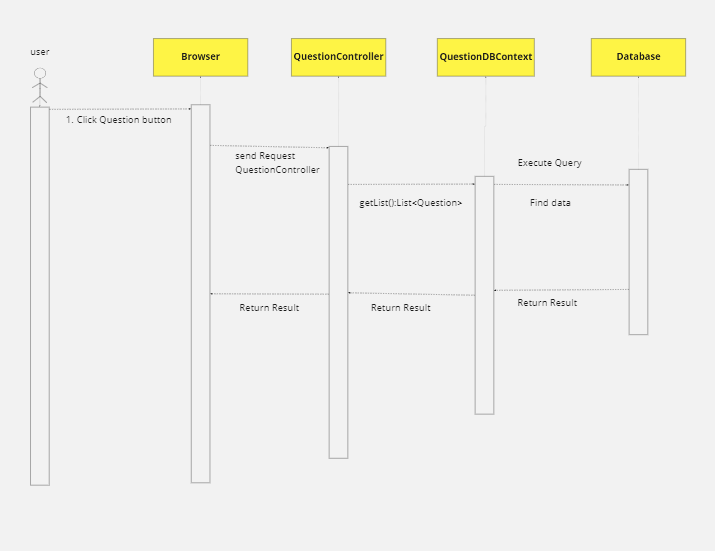
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getQuestionList** | **get all list question have status active()** |
| **2** | **getPagination** | **get all list question pagination** |
| **3** | **searchByTitleAndFiltetMultiFieldAndMultiSort()** | * **User can search Question by content** * **User can sort Question by properties** * **User can filter by Category** |

### c. Sequence Diagram(s)



### d. Database queries

* **getQuestionList:**

****

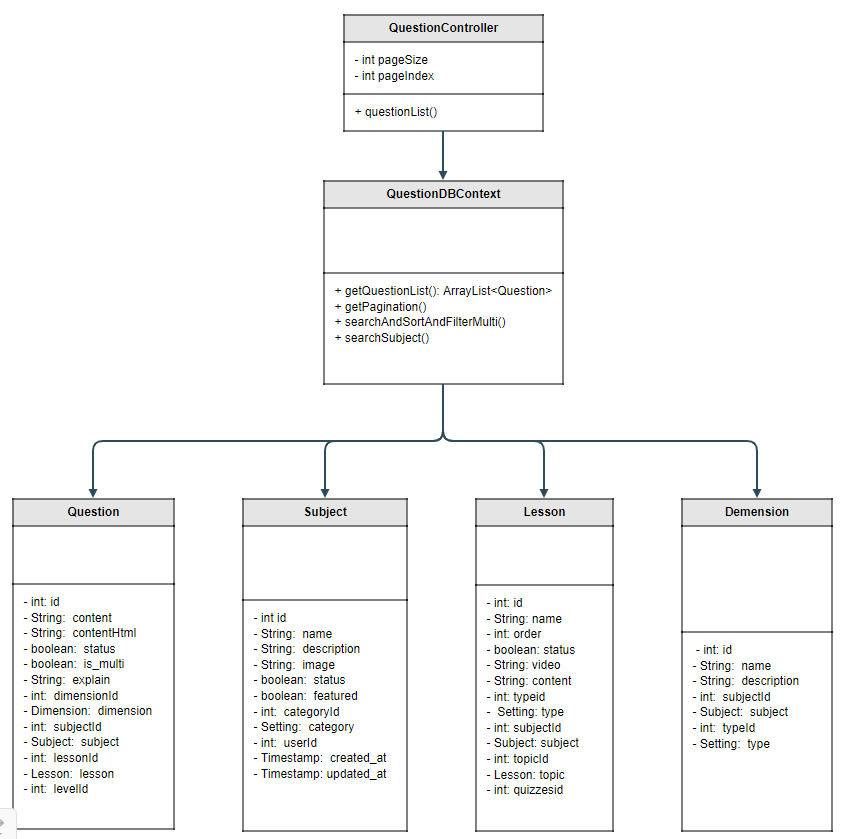
* **searchByTitleAndFiltetMultiFieldAndMultiSort():**

****

### 

## 32. Test Content/Questions Import

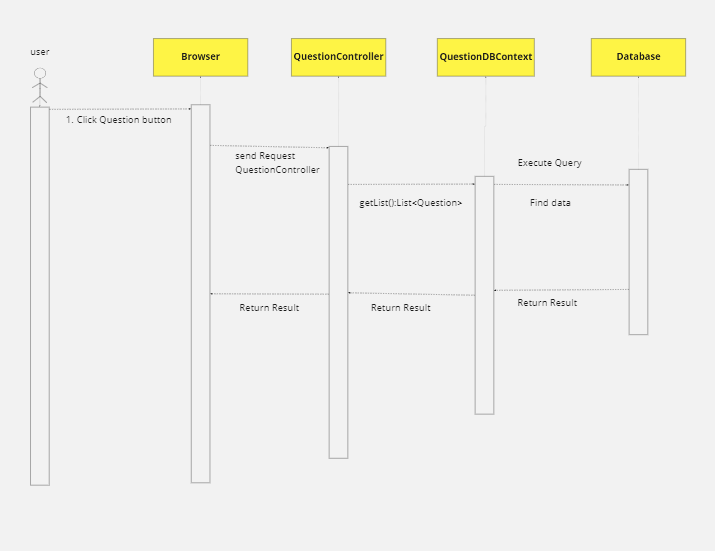
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getQuestionList** | **get all list question have status active()** |
| **2** | **getPagination** | **get all list question pagination** |
| **3** | **searchByTitleAndFiltetMultiFieldAndMultiSort()** | * **User can search Question by content** * **User can sort Question by properties** * **User can filter by Category** |

### c. Sequence Diagram(s)



### d. Database queries

* **getQuestionList:**

****

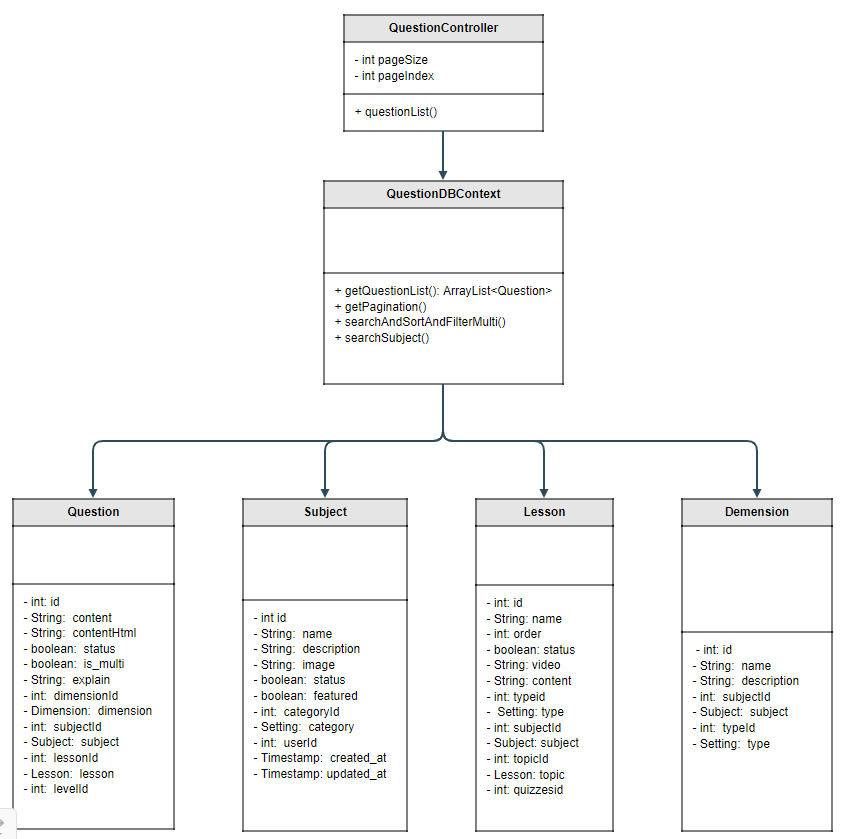
* **searchByTitleAndFiltetMultiFieldAndMultiSort():**

****

### 

## 33. Test Content/ Quizzes List

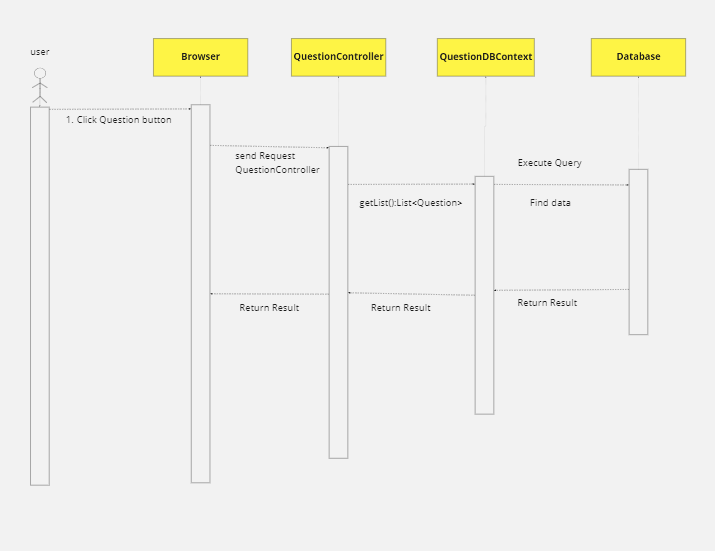
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getQuestionList** | **get all list question have status active()** |
| **2** | **getPagination** | **get all list question pagination** |
| **3** | **searchByTitleAndFiltetMultiFieldAndMultiSort()** | * **User can search Question by content** * **User can sort Question by properties** * **User can filter by Category** |

### c. Sequence Diagram(s)



### d. Database queries

* **getQuestionList:**

****

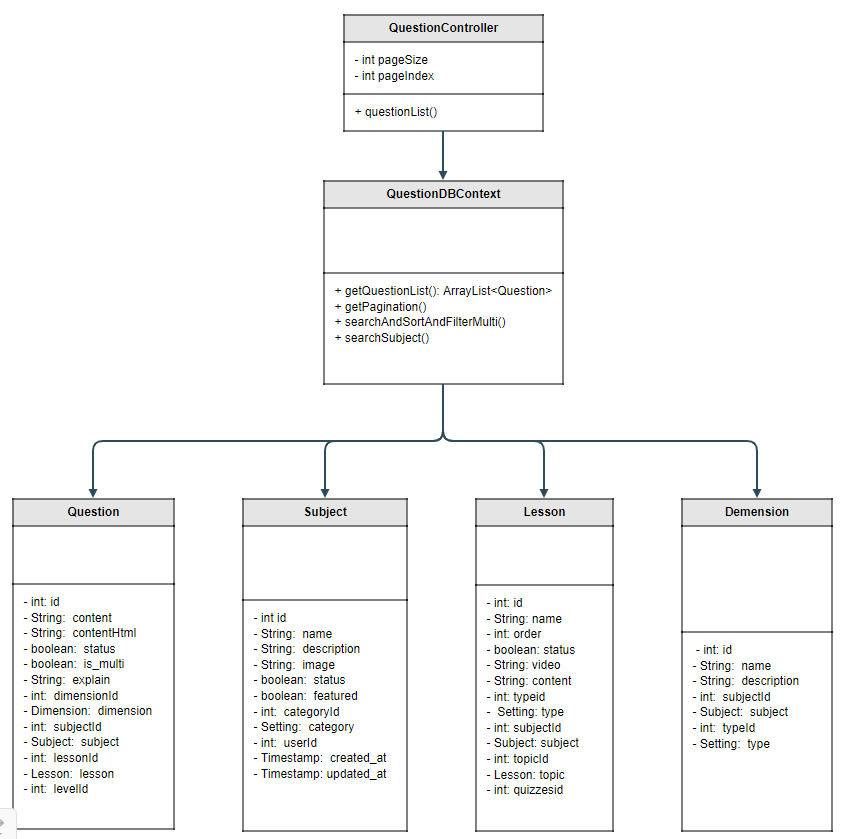
* **searchByTitleAndFiltetMultiFieldAndMultiSort():**

****

### 

## 34. Test Content/ Quizzes Details

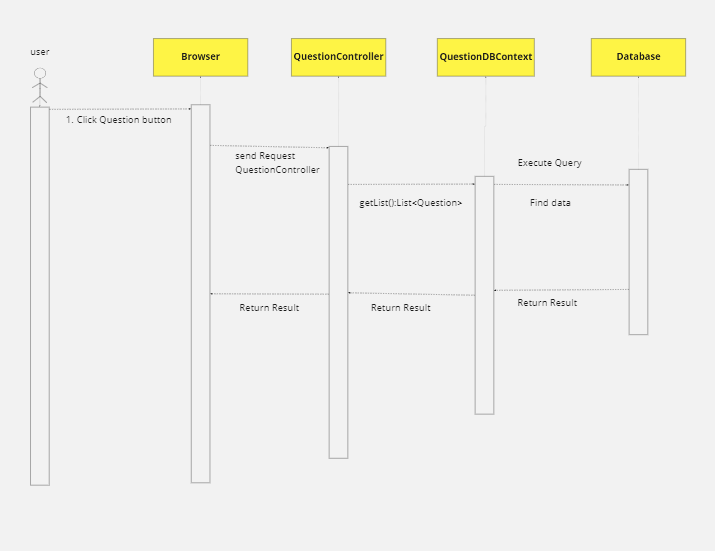
## a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getQuestionList** | **get all list question have status active()** |
| **2** | **getPagination** | **get all list question pagination** |
| **3** | **searchByTitleAndFiltetMultiFieldAndMultiSort()** | * **User can search Question by content** * **User can sort Question by properties** * **User can filter by Category** |

### c. Sequence Diagram(s)



### d. Database queries

* **getQuestionList:**

****

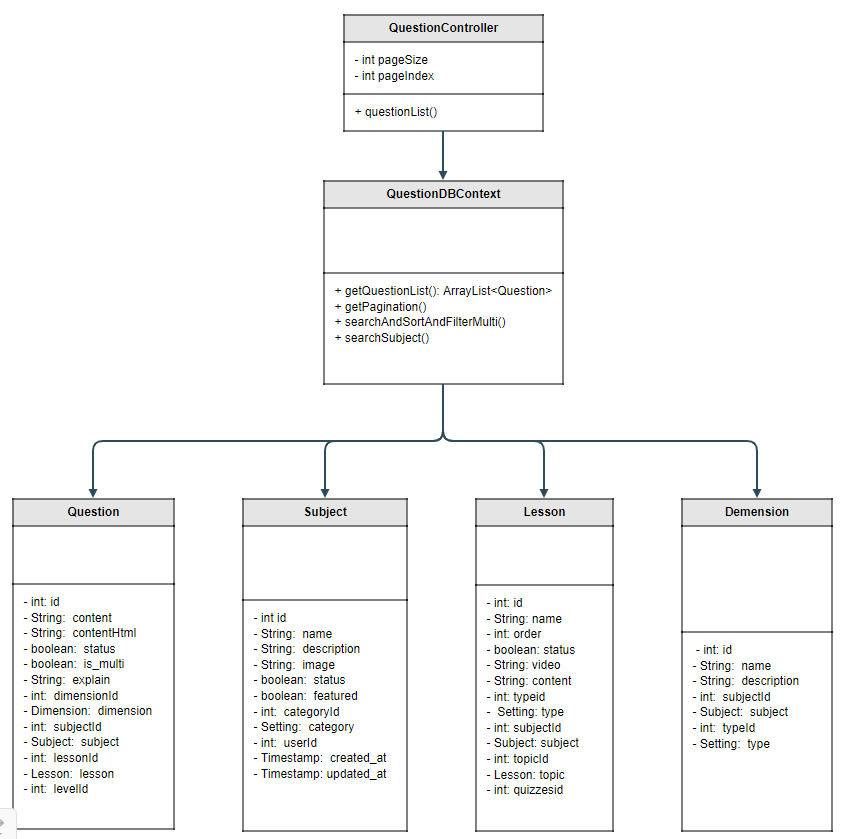
* **searchByTitleAndFiltetMultiFieldAndMultiSort():**

****

## 

## 35. Sale/ Registrations List

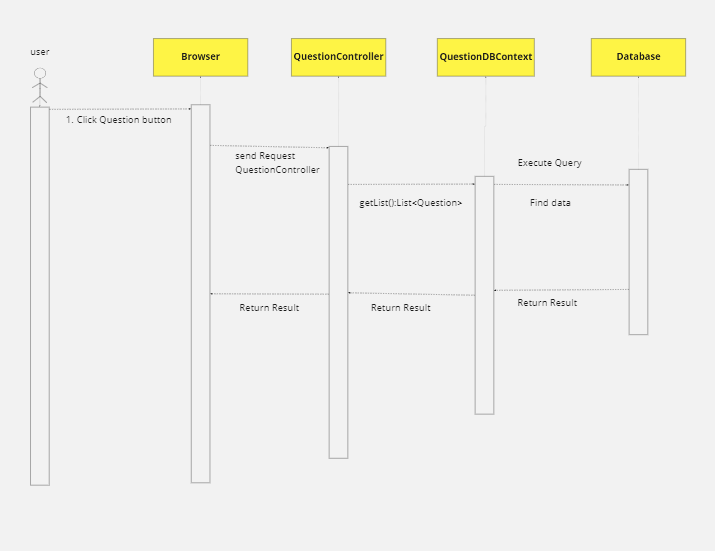
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getQuestionList** | **get all list question have status active()** |
| **2** | **getPagination** | **get all list question pagination** |
| **3** | **searchByTitleAndFiltetMultiFieldAndMultiSort()** | * **User can search Question by content** * **User can sort Question by properties** * **User can filter by Category** |

### c. Sequence Diagram(s)



### d. Database queries

* **getQuestionList:**

****

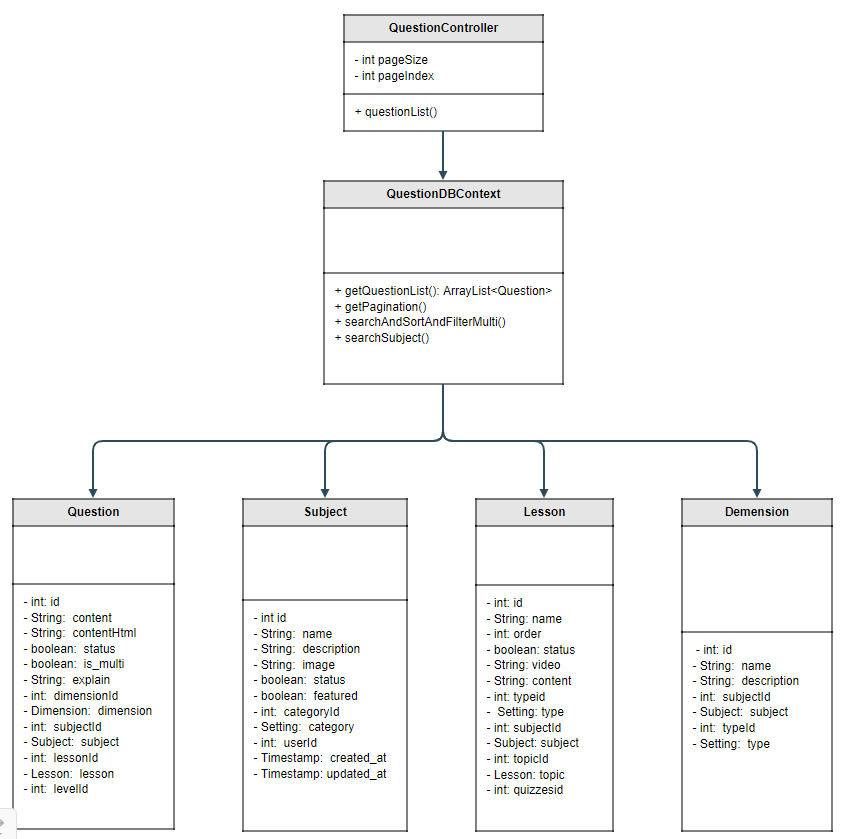
* **searchByTitleAndFiltetMultiFieldAndMultiSort():**

****

### 

## 36. Sale/ Registrations Details

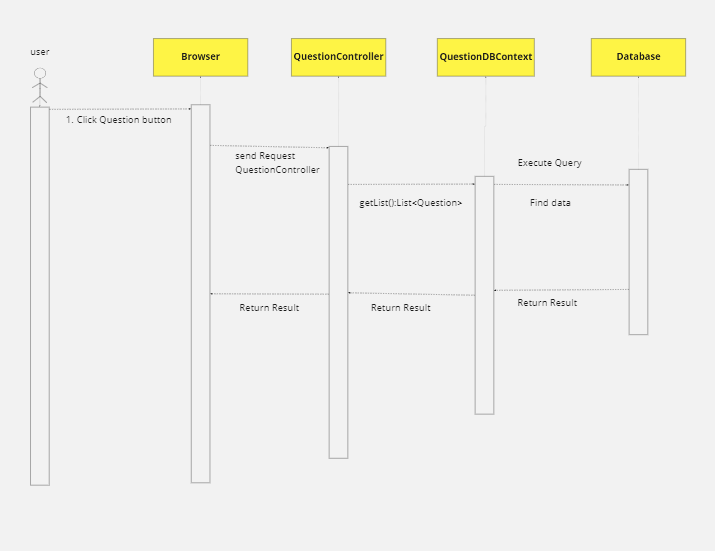
### a. Class Diagram



### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getQuestionList** | **get all list question have status active()** |
| **2** | **getPagination** | **get all list question pagination** |
| **3** | **searchByTitleAndFiltetMultiFieldAndMultiSort()** | * **User can search Question by content** * **User can sort Question by properties** * **User can filter by Category** |

### c. Sequence Diagram(s)



### d. Database queries

* **getQuestionList:**

****

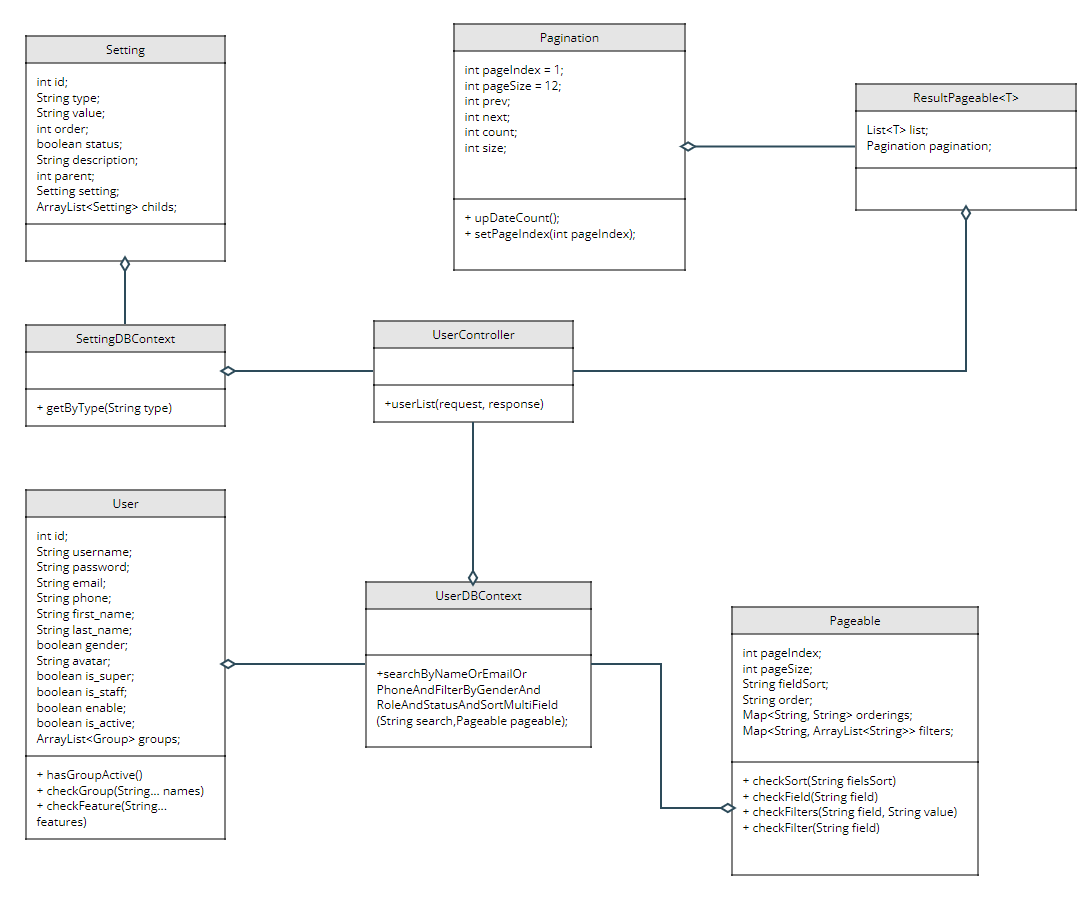
* **searchByTitleAndFiltetMultiFieldAndMultiSort():**

****

### 

## 37. Admin/User List

### a. Class Diagram



### b. Class Specifications

**User:**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **hasGroupActive** | **check user active** |
| **2** | **checkGroup** | **check group** |
| **3** | **checkFeature** | **check feature** |

**UserDBContext:**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **searchByNameOrEmailOrPhoneOrFilterByGenderOrRoleOrStatusOrSortMultiField** | **get list by search name, email, phone or by filter gender, role, status or sort multi field** |

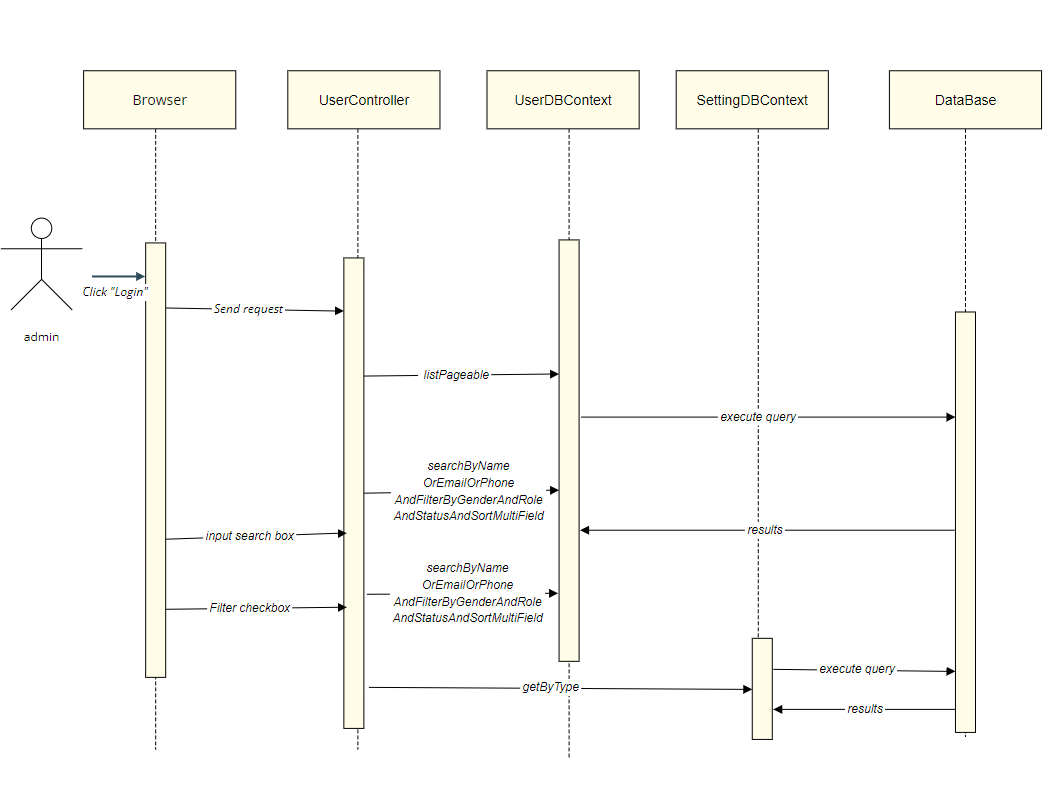
**SettingDBContext:**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getByType** | **get list Setting by type** |

**UserController:**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **userList** | **get request user list** |

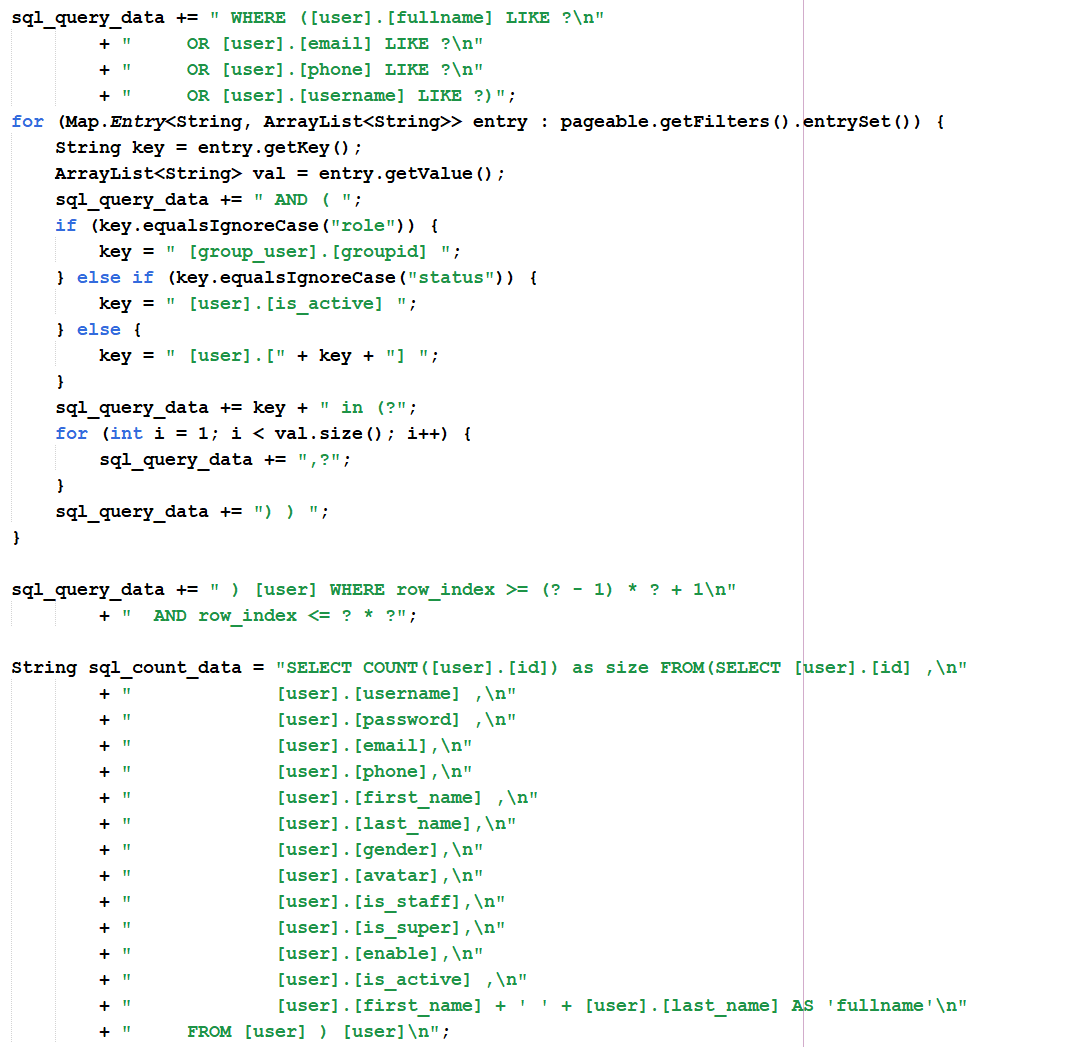
### c. Sequence Diagram(s)

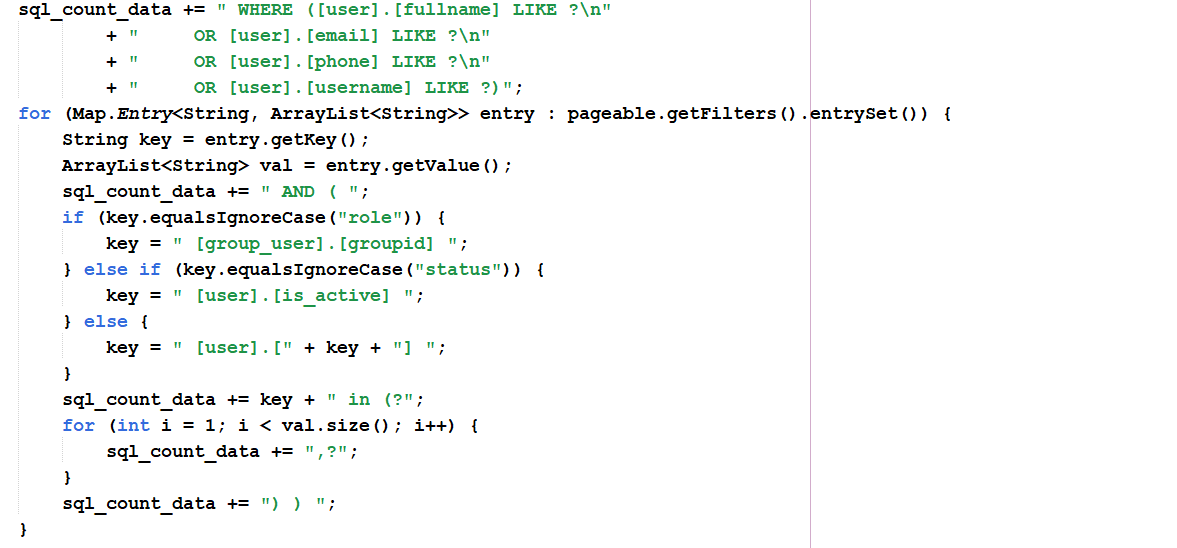


### d. Database queries

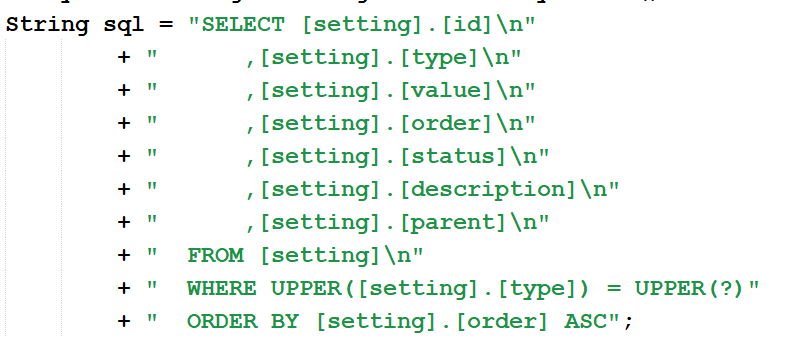
#### Get list user by search Name, Email, Phone Or filter Gender, Role, Status Or Sort MultiField





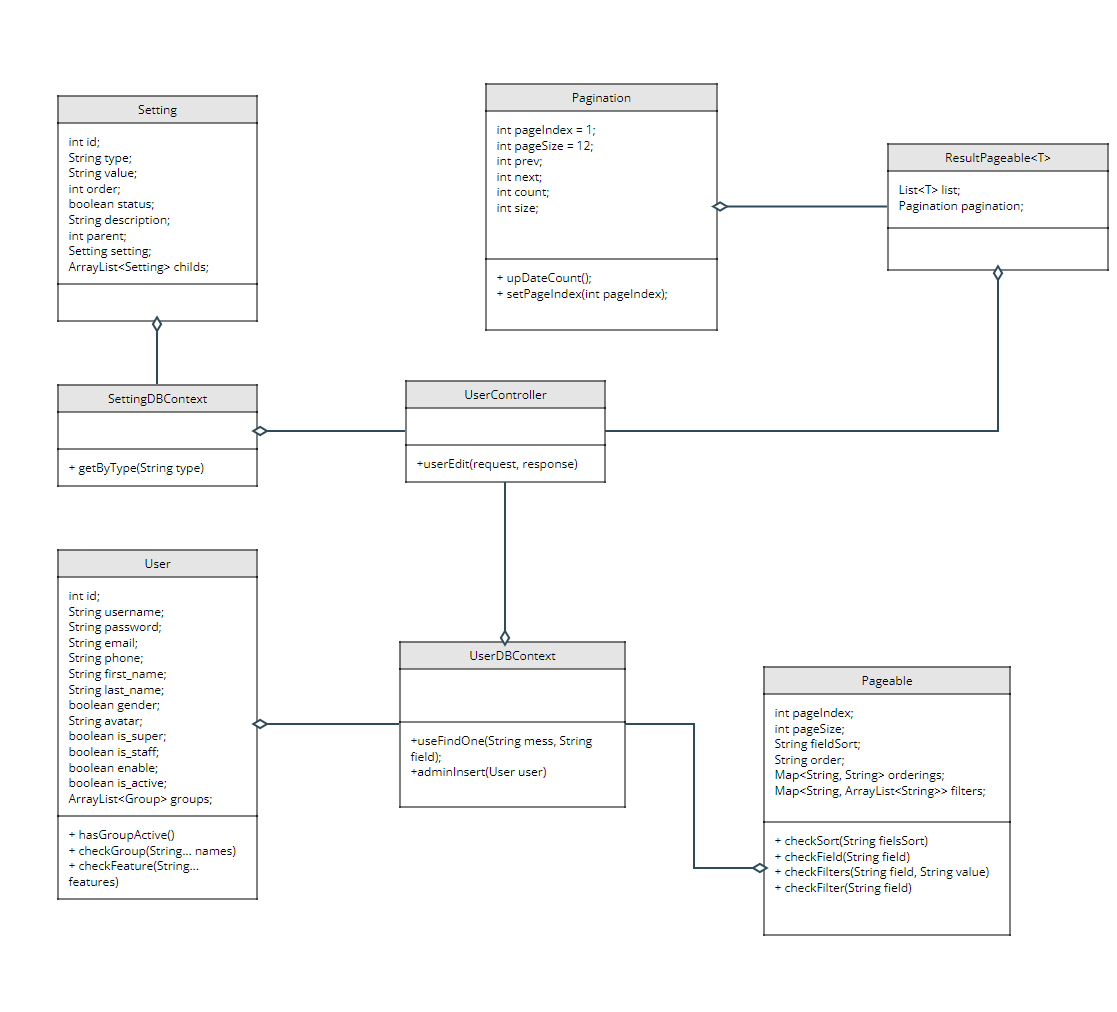


#### Get list setting by type



## 38. Admin/User Details

### a. Class Diagram



### b. Class Specifications

**User:**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **hasGroupActive** | **check user active** |
| **2** | **checkGroup** | **check group** |
| **3** | **checkFeature** | **check feature** |

**UserDBContext:**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **searchByNameOrEmailOrPhoneOrFilterByGenderOrRoleOrStatusOrSortMultiField** | **get list by search name, email, phone or by filter gender, role, status or sort multi field** |

**SettingDBContext:**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getByType** | **get list Setting by type** |

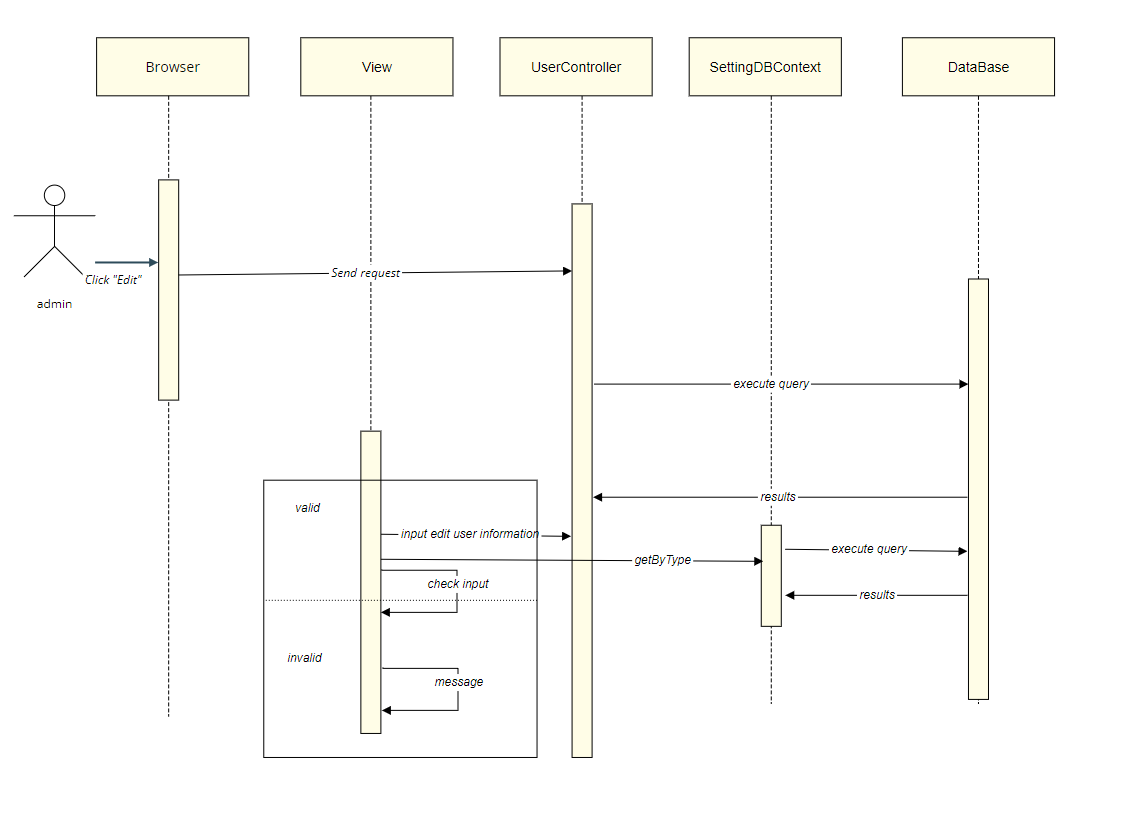
**UserController:**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **userEdit** | **get request user edit** |

**UserDBContext:**

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **userFindOne** | **find user by field** |
| **2** | **adminInsert** | **insert user by admin** |

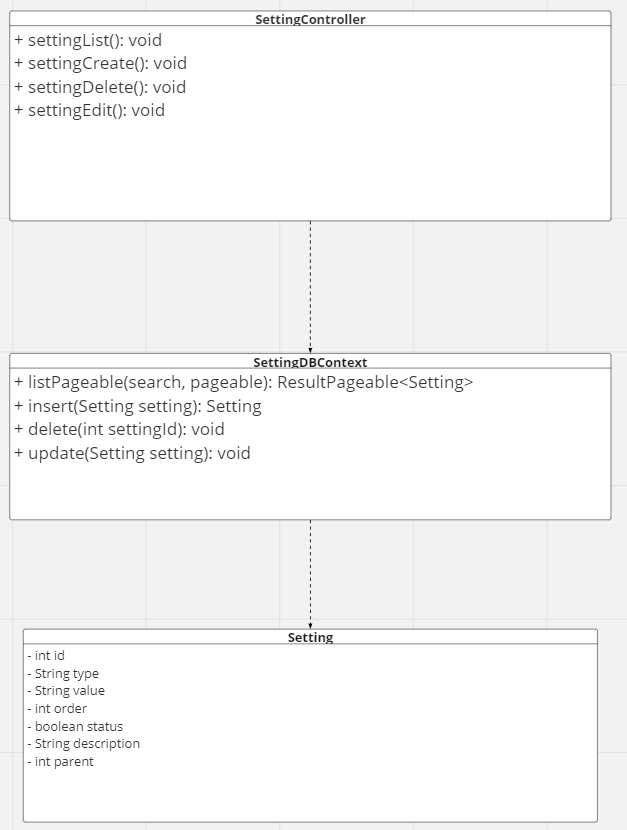
### c. Sequence Diagram(s)



### d. Database queries

## 39. Admin/Settings List

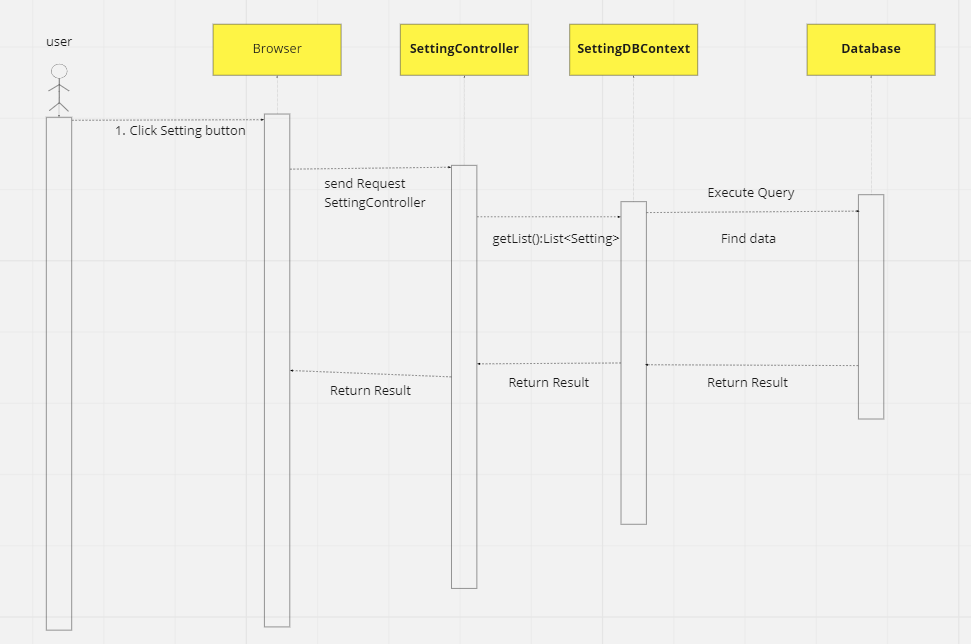
### a. Class Diagram



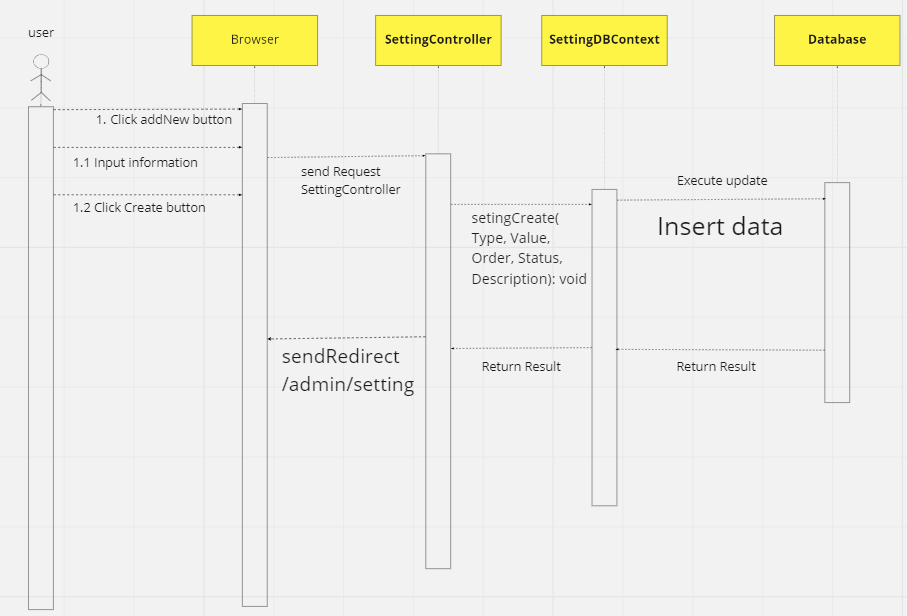
### b. Class Specifications

| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **listPageable()** | * **get all Setting in database** * **User can search setting by title, value** * **User can filter by Types** * **User can sort by id, type, value, parent, order, status** |
| **2** | **insert(Setting setting)** | **use to create new setting** |
| **3** | **delete(int settingId)** | **use to delete setting information** |
| **4** | **update(Setting setting)** | **use to update setting information** |

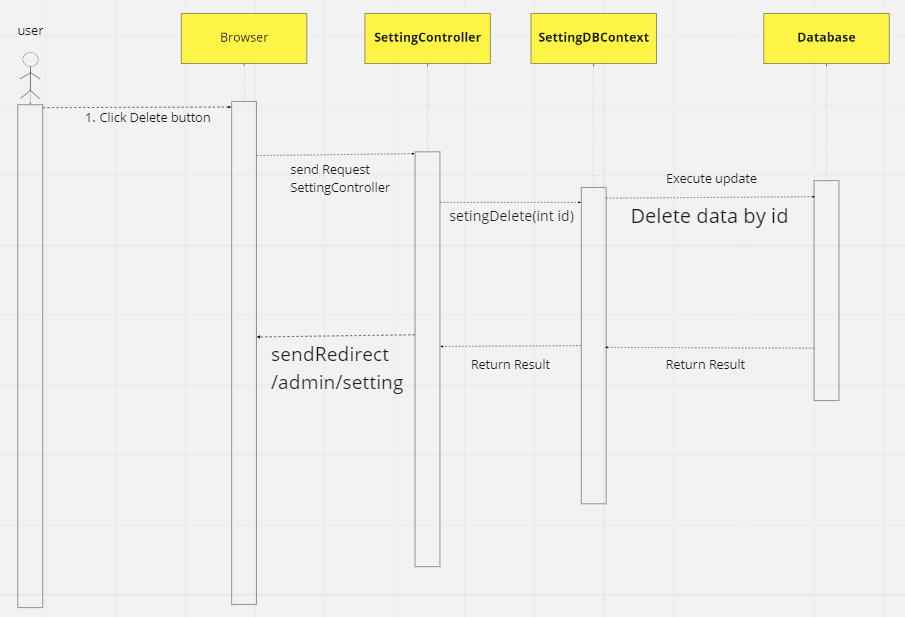
### c. Sequence Diagram(s)



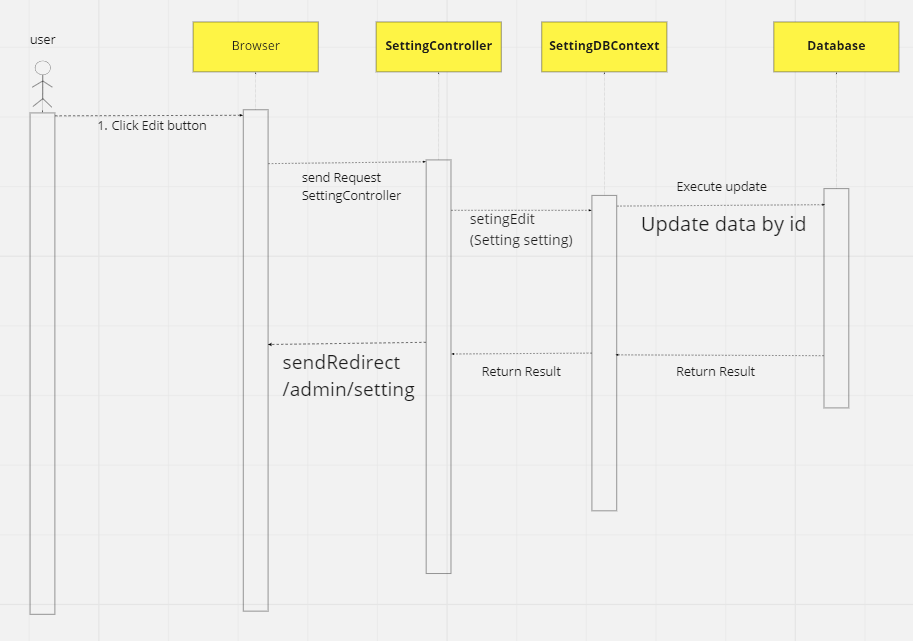
**CREATE NEW SETTING**

****

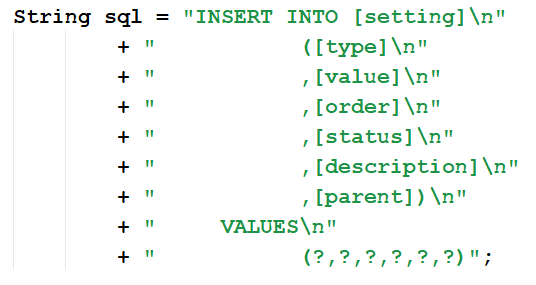
**DELETE SETTING**

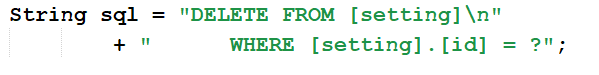
****

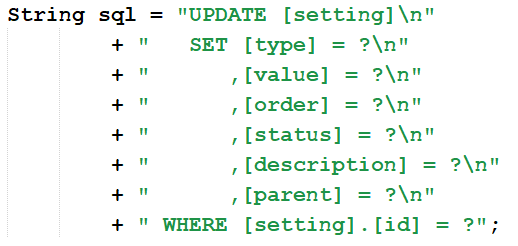
**UPDATE SETTING**

****

### d. Database queries

****

****

****

## 40. Admin/Settings Details

### a. Class Diagram

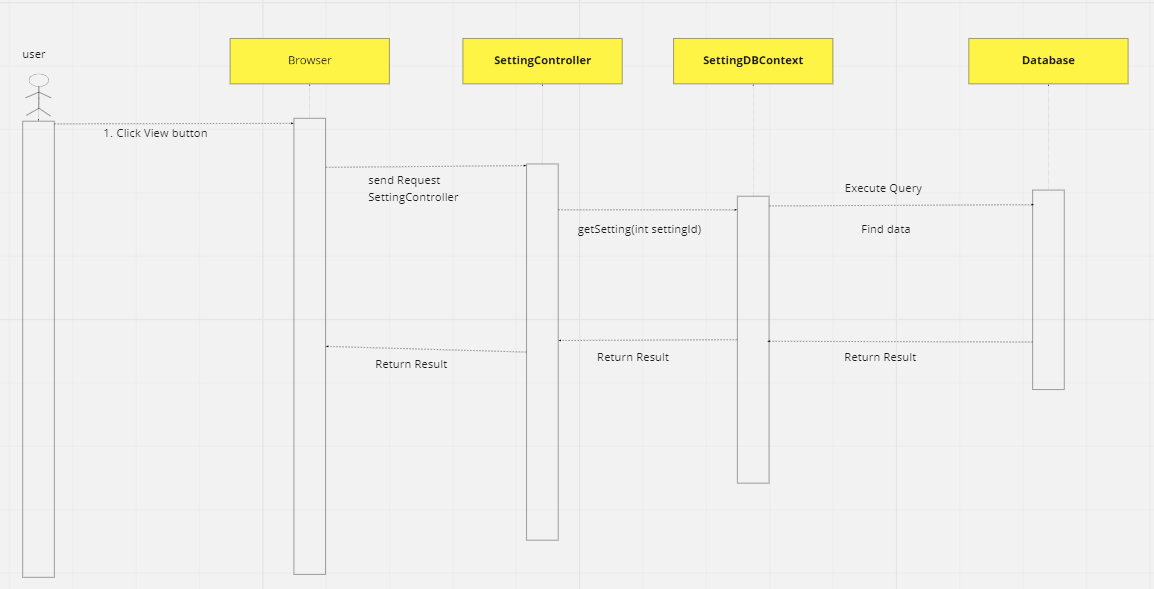


### b. Class Specifications

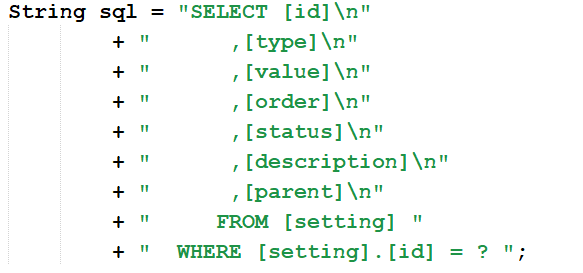
| **No** | **Method** | **Description** |
| --- | --- | --- |
| **1** | **getSetting(int settingId)** | **get 1 setting by id** |

### 

### c. Sequence Diagram(s)

****

### d. Database queries

****

# III. Database Design

## 1. Database Schema

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

## 2. Table Description

| **No** | **Table** | **Description** |
| --- | --- | --- |
| *01* | *<Table name>* | *<Description of the table>*  *- Primary keys: <<list of primary key fields>>*  *- Foreign keys: <<list of foreign key fields>>* |
| *02* |  |  |