

**YOGA CENTER**

**Software Design Specification**

– HoChiMinh, July 2023 –

**Record of changeS**

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| July 12 | A | AnLH | Add detail Code Designs 3, 4, 5 |
| July 12 | A | DungVNT | Code package added, |
| July 15 | A | ThangNC | Code package modified and add description |
| July 15 | A | ThangNC | Add details to section II Code Designs item number 8, 9, 10 |
| July 14 | A | LongNT | Modified details to section II Code Designs - Sequence diagram of item 6, 7 |
| July 15 | M | AnLH | Add Code Designs (11, 12, 13, 14) |
| July 15 | M | DatTGT | Modified table description (13, 18, 19, 20) |
| July 17 | A | ThangNC | Add Search, Category. HomePage function |
| July 17 | A | DatTGT | Double check & edit errors |
| July 18 | A | DungVNT | Add details to section II Code Designs item number 1, 6, 9, 15 |
| July 19 | A | AnLH | Add description of table in database design |
| July 20 | A | LongNT, DungVNT | Check All |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview 5](#_gjdgxs)

[1. Code Packages 5](#_30j0zll)

[2. Database Design 6](#_1fob9te)

[a. Database Schema 6](#_3znysh7)

[b. Table Description 6](#_2et92p0)

[3. Usecase 7](#_b9dp6a1h7vjn)

[4. ERD Diagram 8](#_wecxhj96swwf)

[II. Code Designs 9](#_tyjcwt)

[1. Payment course 9](#_3dy6vkm)

[a. Class Diagram 9](#_1t3h5sf)

[b. Class Specifications 9](#_4d34og8)

[c. Sequence Diagram(s) 11](#_2s8eyo1)

[d. Database Queries 11](#_17dp8vu)

[2. View Blog 12](#_3rdcrjn)

[a. Class Diagram 12](#_sss7ze7zwzbj)

[b. Class Specifications 12](#_7mss6v7nu3gf)

[c. Sequence Diagram(s) 14](#_721xy355mrm7)

[d. Database Queries 14](#_9jtuso7s03zm)

[3. Manage Account 15](#_1tftdptwu3lp)

[a. Class Diagram 15](#_1ozhge3y15oy)

[b. Class Specifications 15](#_6qvq03i5d3a7)

[c. Sequence Diagram(s) 17](#_rb83t1pc9ady)

[d. Database Queries 17](#_4oq14zo4sjkr)

[4. View Timetable 18](#_booneufvtru5)

[a. Class Diagram 18](#_1w489vc5pd51)

[b. Class Specifications 18](#_sk2rv6i814vx)

[c. Sequence Diagram(s) 19](#_5r7c3nhm9y8)

[5. Booking course 21](#_rac58jsq5i3i)

[a. Class Diagram 21](#_uopzxaew3k69)

[b. Class Specifications 21](#_v6gyi5jr5zpl)

[AccountRepository Class 21](#_cyoz5asjg53g)

[c. Sequence Diagram(s) 24](#_460jdc55179i)

[d. Database Queries 24](#_7pk8een3sgkp)

[6. Bill Analysis 26](#_jshfyto4d9jr)

[a. Class Diagram 26](#_pf0v3rjyhk49)

[b. Class Specifications 26](#_fg6p0osqr882)

[c. Sequence Diagram(s) 28](#_pc9sihwgj0xp)

[d. Database Queries 28](#_45znm43hi21v)

[7. Rating course 29](#_ce7kyq6wsidm)

[a. Class Diagram 29](#_ijtr0p66cmou)

[b. Class Specifications 29](#_sj5aiokygy7k)

[AccountRepository Class 30](#_pmq4g6glgqeu)

[c. Sequence Diagram(s) 32](#_8gzernva83d9)

[d. Database Queries 32](#_nq58yb5cic6c)

[8. Cashier ViewBill 33](#_dc6ti87rwa15)

[a. Class Diagram 33](#_8n9kxpesskwy)

[b. Class Specifications 33](#_iwqxxzep1jjc)

[c. Sequence Diagram(s) 35](#_f2y9ntgnbnyh)

[d. Database Queries 35](#_rcb65w7c7hwu)

[9. Blog Analysis 36](#_rla7muryzrh4)

[a. Class Diagram 36](#_1390bxvcm6hb)

[b. Class Specifications 36](#_yjoc8q17jj3k)

[c. Sequence Diagram(s) 38](#_kk96yeid50h3)

[d. Database Queries 38](#_rvf67x413y9d)

[10. Cashier UpdateBill 39](#_6hs2q2p4nenr)

[a. Class Diagram 39](#_ju1huglahlq0)

[b. Class Specifications 39](#_46y4lng88ahm)

[c. Sequence Diagram(s) 41](#_gf47w0p7tl9g)

[d. Database Queries 41](#_3m0t939yc2ns)

[11. Update User Profile 42](#_e3shqwdq3bwo)

[a. Class Diagram 42](#_56oxc7nm8jrc)

[b. Class Specifications 42](#_t5sagywh7g5p)

[AccountRepository Class 42](#_dxdzekk2c6g7)

[c. Sequence Diagram(s) 43](#_owjzlc9bhxfn)

[d. Database Queries 43](#_u1ux1pek3e0g)

[12. View Timetable List 44](#_u5td6bggki55)

[a. Class Diagram 44](#_j8ig1sf4b974)

[b. Class Specifications 44](#_doyddqc04r7)

[c. Sequence Diagram(s) 45](#_coa48x7s14wv)

[d. Database Queries 45](#_gxuxocydly7n)

[13. Buy membership 46](#_3qigcld61ko)

[a. Class Diagram 46](#_yu3b45o576z6)

[b. Class Specifications 46](#_nua1i0ba445t)

[c. Sequence Diagram(s) 48](#_rm4ghn6ylkk5)

[d. Database Queries 48](#_yoh3hmfejy0i)

[14. View membership 49](#_mjnt1a1mrzs1)

[a. Class Diagram 49](#_661468gtv6e9)

[b. Class Specifications 50](#_b6qajkwgu94s)

[c. Sequence Diagram(s) 52](#_b3yr9lch9g9a)

[d. Database Queries 52](#_wrmuz7d8ubms)

[15. View Bill 53](#_4u08l8mi43nv)

[a. Class Diagram 53](#_offyl2a9isvc)

[b. Class Specifications 54](#_yenjl36hjo8c)

[c. Sequence Diagram(s) 54](#_duo7ni1ykykp)

[d. Database Queries 55](#_jiaebr6fezq0)

[16. View Course Time List 55](#_27cols7kpked)

[a. Class Diagram 55](#_1io9pz7fp4be)

[b. Class Specifications 56](#_syam28gy605)

[c. Sequence Diagram(s) 57](#_rp95trhljq27)

[d. Database Queries 57](#_jz0l8kru0lis)

[17. Cancel Booking 58](#_r06p4uors0s)

[a. Class Diagram 58](#_ia4iws9ni68r)

[b. Class Specifications 58](#_8gmhc1pklthv)

[AccountRepository Class 58](#_qdwzil9as4yx)

[BillRepository Class 59](#_so3tzl2ww4w)

[c. Sequence Diagram(s) 60](#_l1g7rh1b0218)

[d. Database Queries 60](#_libjstj7op1s)

[18. Cashier ViewBillMembership 61](#_pvnhif55vcli)

[a. Class Diagram 61](#_ybv2q75n3efk)

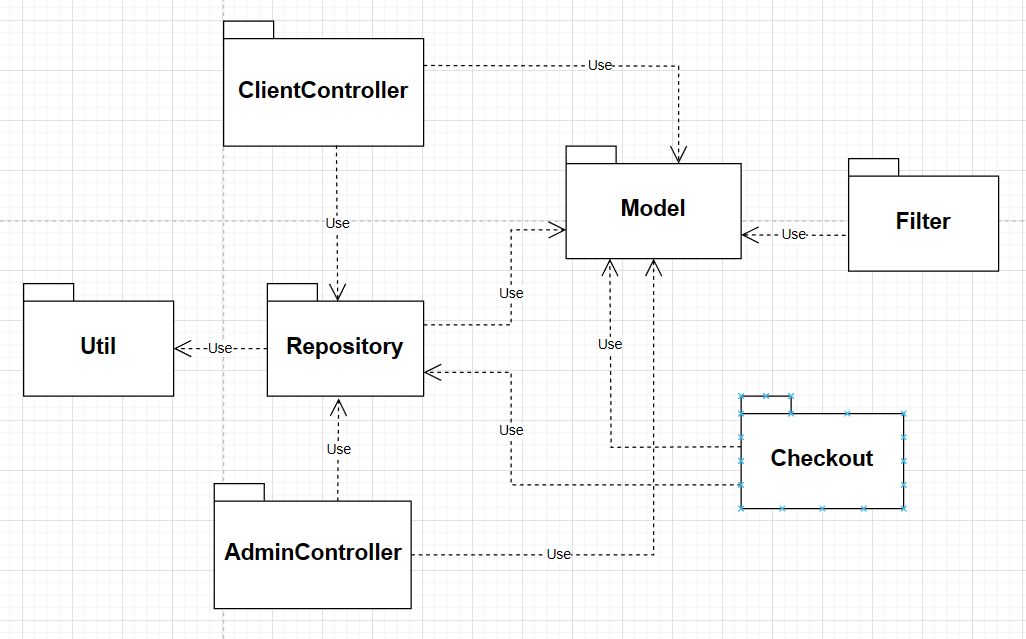
[b. Class Specifications 61](#_izq1cecyyc8a)

[c. Sequence Diagram(s) 62](#_s46a8jbek8a0)

[d. Database Queries 62](#_vag0hk2mmmxc)

# I. Overview

## 1. Code Packages

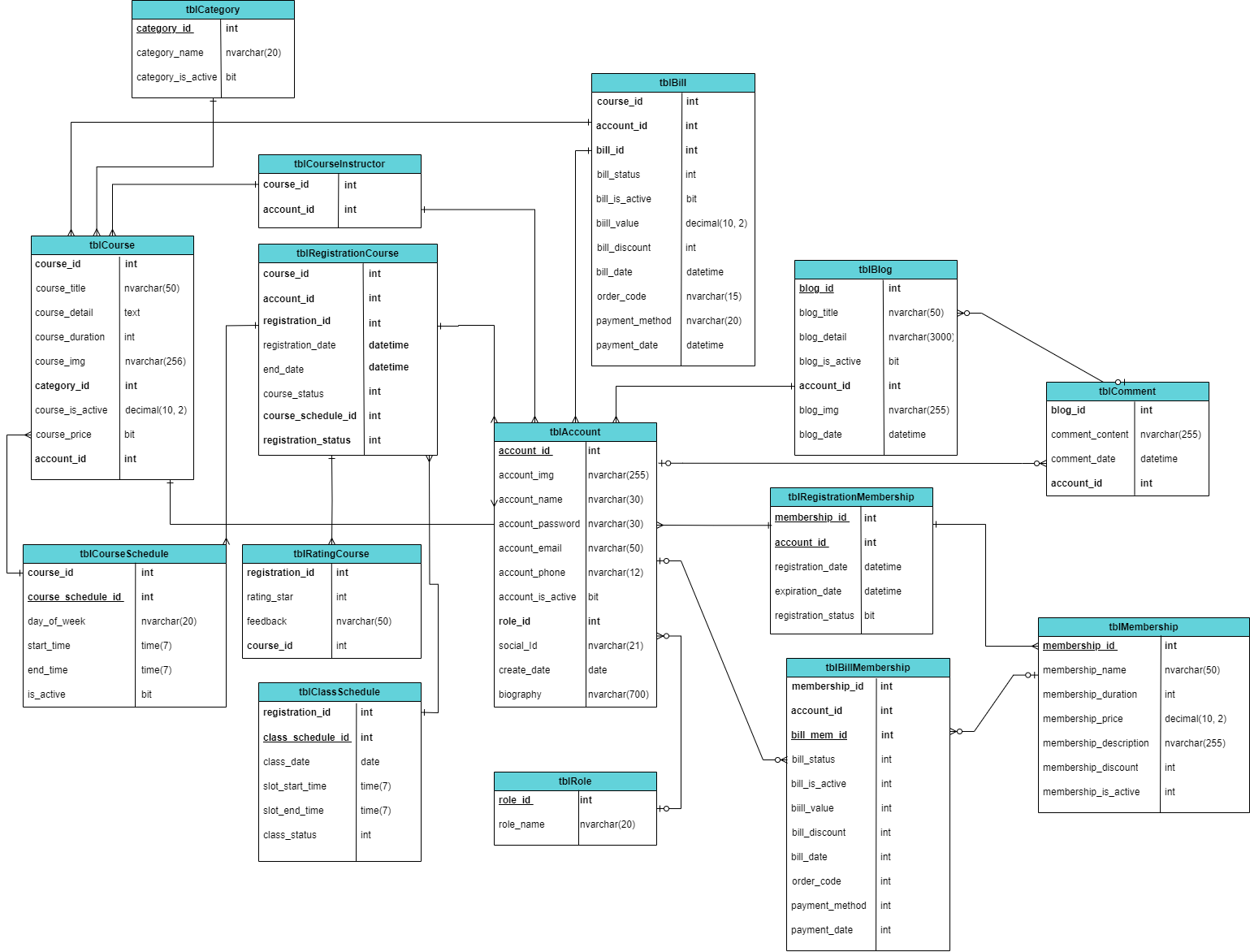


***Package descriptions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| 01 | Util | Contains various utility classes and components that provide generic functionalities and services that can be used throughout the application. |
| 02 | Repository | Provides an interface for accessing and manipulating data from the underlying data source. |
| 03 | Model | Defines the domain models that represent the entities in the application. |
| 04 | ClientController | Handles client interactions and invokes appropriate actions. |
| 05 | AdminController | Handles admin interactions and invokes appropriate actions. |
| 06 | Checkout | Handles client interactions in checkout progress and interacts with a payment gateway to generate bills and checkout. |
| 07 | Filter | Restrict users to interact directly with the website via URL. Handle authentication user. |

## 2. Database Design

### a. Database Schema



### b. Table Description

| **No** | **Table** | **Description** |
| --- | --- | --- |
| 01 | tblCategory | tblCategory is related to category\_id, category\_name, and category\_is\_active. Categories are used to represent the type of yoga of the course. |
| 02 | tblCourse | tblCourse will include information such as name, time, type, price, participants, status, and an image representing the course. The course will allow users to buy and participate. |
| 03 | tblCourseSchedule | Course Schedule - specific attributes include the date of the week, slot start time, slot end time, and is active. |
| 04 | tblCourseInstructor | tblCourseInstructor is composed of account\_id and course\_id. This table will determine which account will be the trainer of which course. |
| 05 | tblRegistrationCourse | tblRegistrationCourse will include course\_id, account\_id, registration\_id, registration\_date , end\_date, course\_status , course\_schedule\_id , registration\_status. This table will store the account's course registration information. |
| 06 | tblClassSchedule | Class Schedule - specific attributes include class date, slot start time, slot end time, and status. |
| 07 | tblRatingCourse | Store data with feedback, rating star, and course ID which course users register |
| 08 | tblComment | It is a comment from the user it is a possible vision all viewer |
| 09 | tblMembership | Set of "Membership" in the system which provides a lot of discount for the trainee, and benefits at the center when becoming a member. |
| 10 | tblBill | Store information user who has paid for Course with attributes including the Account ID, Course ID, Bill ID, Status, Price, Discount, Payment Date, Payment Method and Order Code |
| 11 | tblBlog | Blog - specific attributes include blog detail, title, create date,e.g |
| 12 | tblRegistrationMemembership | tblRegistrationMemembership will include membership\_id, account\_id, registration\_date, expiration\_date, registration\_status. This table will store the membership card information of the account when the account registers to buy or update. |
| 13 | tblBillMembership | Store information user who has paid for membership with attributes including Account ID, Membership ID, Bill Membership ID, Status, Price, Discount, Payment Date, and Order Code. |
| 14 | tblRole | Role-related information such as role ID and role name. Roles are used to group users with similar permissions or access levels. |
| 15 | tblAccount | Account-specific information such as account ID, username, email, password, and other user-related properties. |

## 

## 

## 

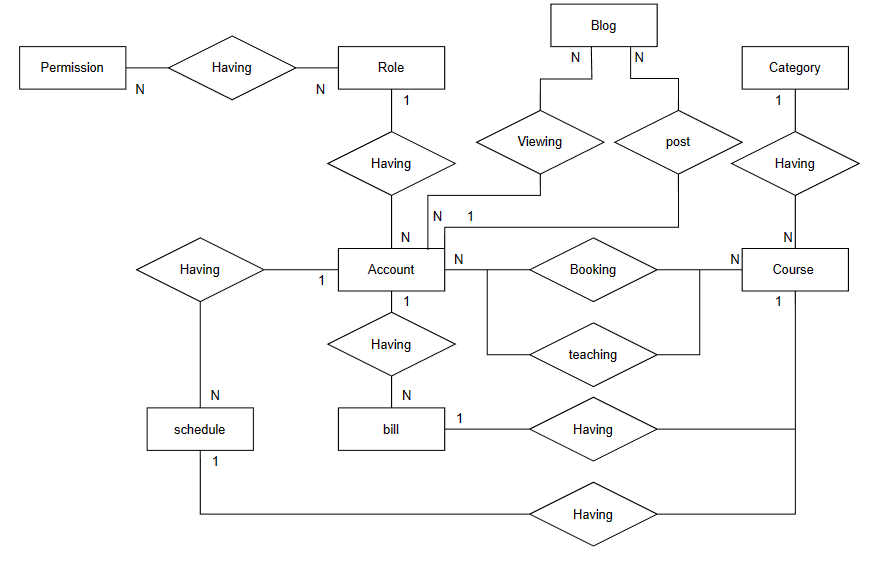
## 3. Usecase

# 

## 

## 

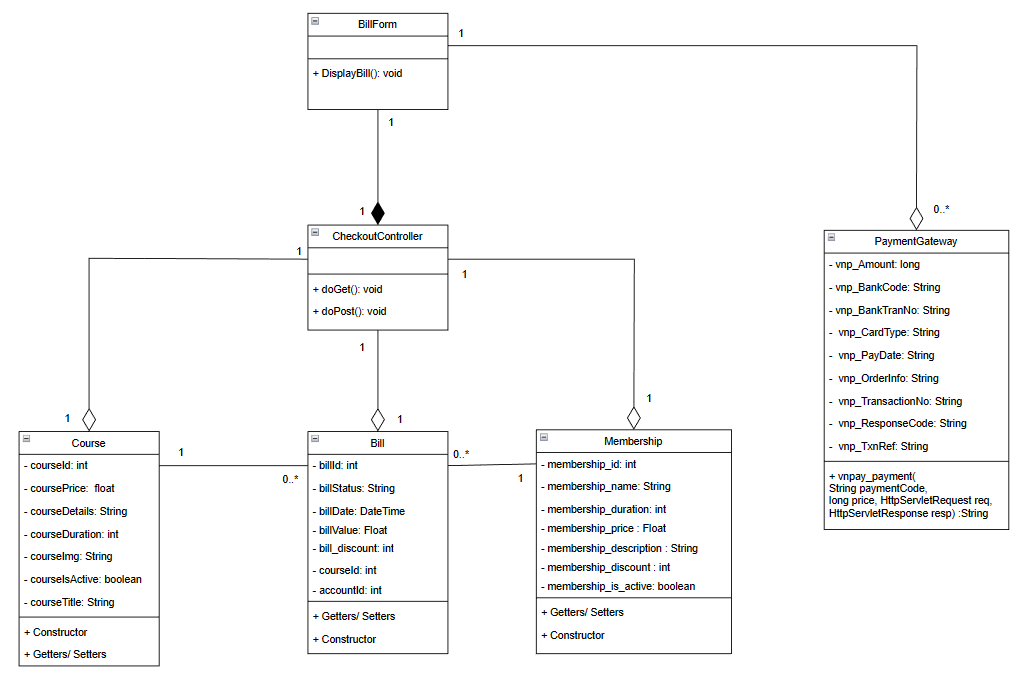
## 4. ERD Diagram



# II. Code Designs

## 1. Payment course

### a. Class Diagram



### b. Class Specifications

#### Course Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter | Get value in attributes |
| 02 | Setter | Set value for attributes |
| 03 | Constructor | Generate Course object |

#### Bill Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter | Get value in attributes |
| 02 | Setter | Set value for attributes |
| 03 | Constructor | Generate Bill object |

#### Membership Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter | Get value in attributes |
| 02 | Setter | Set value for attributes |
| 03 | Constructor | Generate Membership object |

#### CheckoutController Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doget() | Input payment method, generate and get oder code back and store to database |
| 02 | doPost() | Input payment method, generate and get oder code back and store to database |

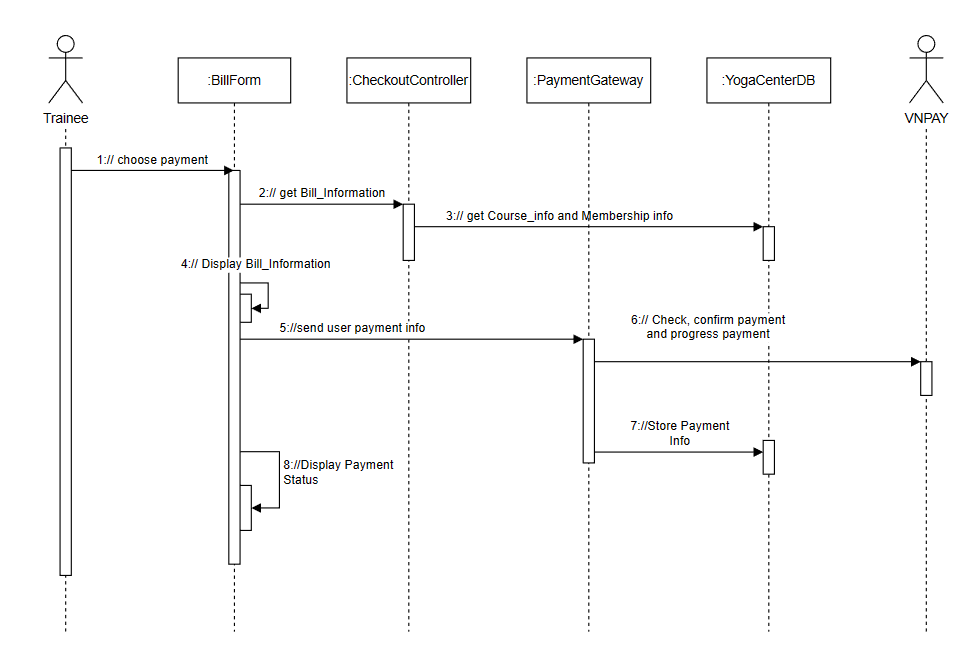
#### BillForm Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | DisplayBill() | Use jsp, jstl, css to generate view for checkout |

#### PaymentGateway Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | vnpay\_payment(..) | Send information of the bill to payment gateway and get back attribute to store in database |

### c. Sequence Diagram(s)



### d. Database Queries

SELECT \* FROM TblBill where id =?

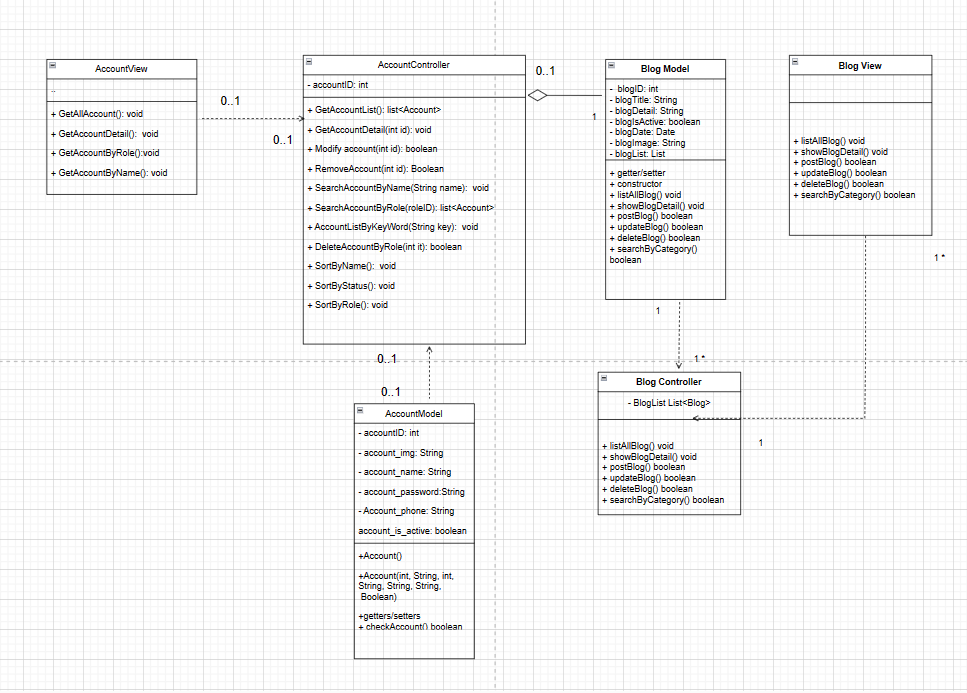
SELECT \* FROM TblCourse where id =?

SELECT \* FROM TblMemberShip where id =?

INSERT INTO tblBill (course\_id, account\_id, bill\_status, bill\_is\_active, bill\_value, bill\_discount, bill\_date, order\_code, payment\_method) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)

## 2. View Blog

### a. Class Diagram



### b. Class Specifications

#### BlogModel Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Blog() | The constructor for class |
| 02 | setter() | Method to update attribute for class, input is the data type of attribute, the output is void |
| 03 | getter() | Method to get the value of an attribute in class, out is value off attribute, input is none |

#### BlogRepository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | List<Blog> getAll | Method get all blog from database |
| 02 | Blog detail(int id) | Method get detail blog where blog\_id =? |
| 03 | update(Blog c) | Method update blog\_title , blog\_detail, blog\_is\_active, account\_id, blog\_img, blog\_date where blog\_id = ? |
| 04 | delete(int id) | Method update blog\_is\_active where blog\_id = ? |
| 05 | count() | Get number of blog |
| 06 | create(Blog blog) | Add new blog |
| 07 | getRecentBlog(int offset,int next,int accountId) | Get all blog where account\_id = ? |
| 08 | getTotalBlog(int accountId) | Get count of blog where account\_id = ? |

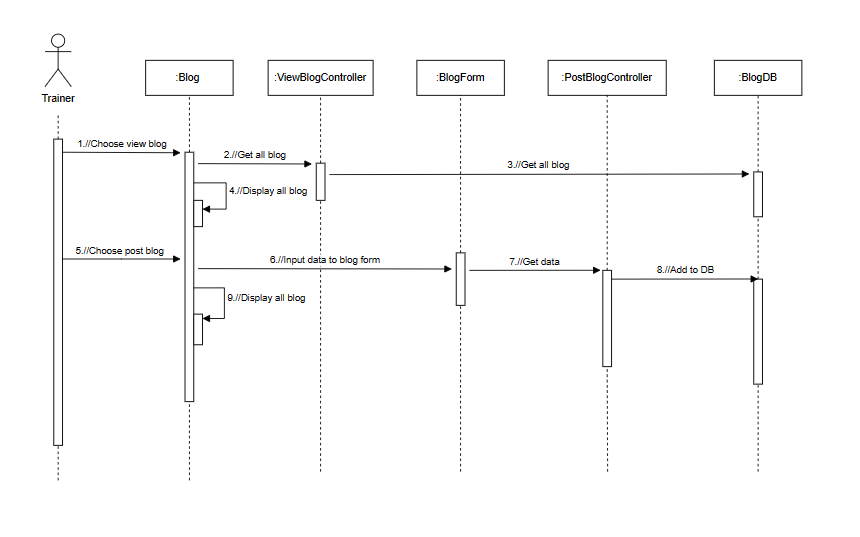
#### 

#### BlogHomeController Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doGet() | Method get all blog recent and blog pagination for blog |
| 02 | doPost() | Get data from user when post blog, edit the blog and delete blog |
|  |  |  |
|  |  |  |

#### 

### c. Sequence Diagram(s)



### d. Database Queries

select \* from tblBlog order by blog\_date desc OFFSET? ROWS FETCH NEXT? ROWS ONLY

select count(\*) as num from tblBlog

INSERT INTO tblBlog ("

+ "blog\_img, blog\_date, account\_id, "

+ "blog\_detail, blog\_title) "

+ "VALUES (?, ?, ?, ?, ?)"

select \* from tblBlog where account\_id=? order by blog\_date DESC OFFSET? ROWS FETCH NEXT? ROWS ONLY

select COUNT(\*) as count from tblBlog where account\_id=?

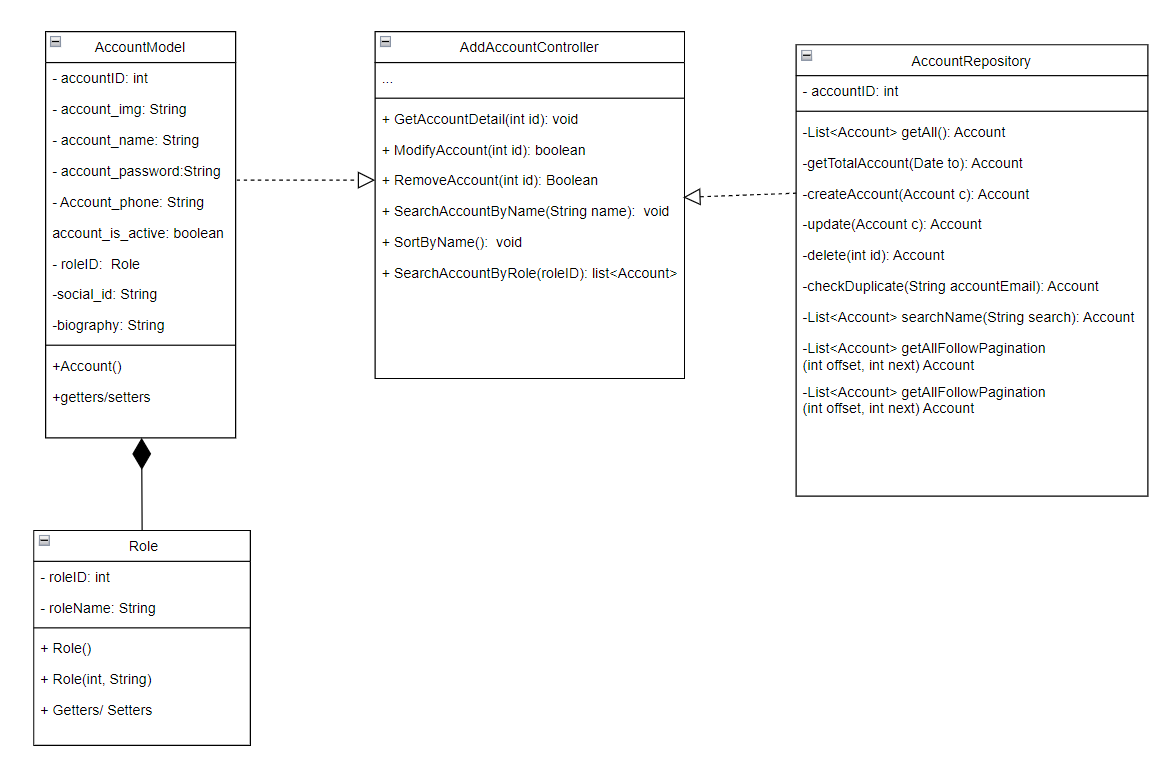
select \* from tblBlog where blog\_id =?

update tblBlog set blog\_title=? , blog\_detail=? , account\_id=? , blog\_date=? , blog\_img=? where blog\_id=?

delete from tblBlog where blog\_id=?

## 3. Manage Account

### a. Class Diagram



### b. Class Specifications

Description: Display the account information that has joined the system and performed the functions on the account.

#### AccountModel Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Account() | Constructor for class. |
| 02 | setter() | Method to set the date in class. |
| 03 | getter() | Method to get the date from class. |

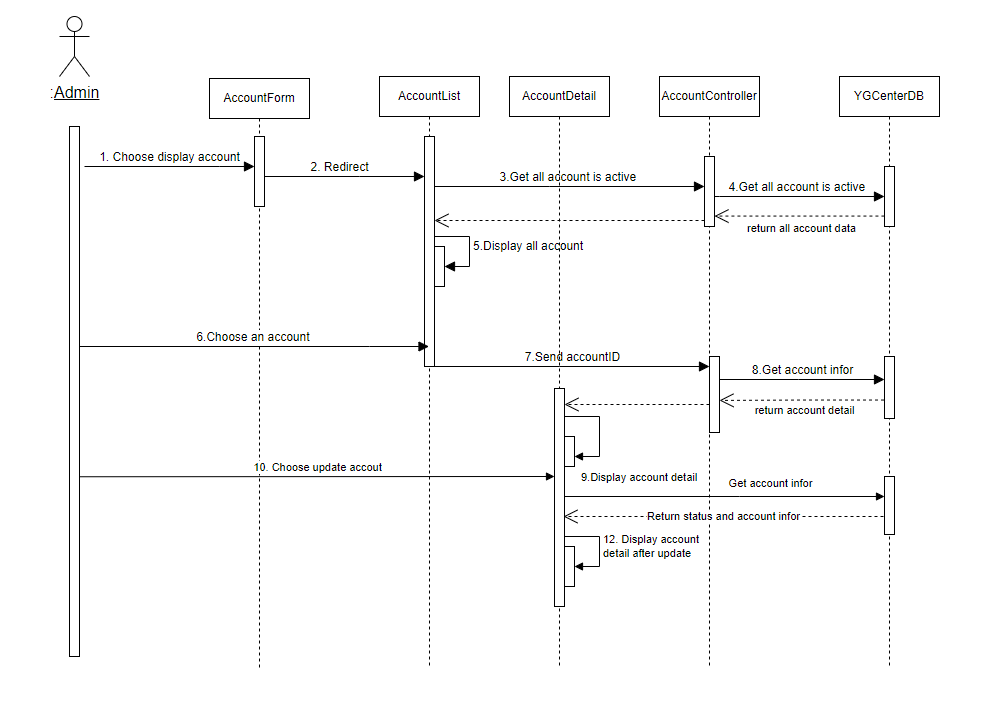
***Account Repository***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | List<Account> getAll() | The function used to query data, get an account list from the database |
| 02 | getTotalAccount() | The function used to query data, get an account list by date nearest to the database |
| 03 | updateAccount() | The function used to query data, edit data of account into the database |
| 04 | createAccount() | The function used to query data, insert new account into the database |
| 06 | deleteAccount() | The function used to query data, update status of the account into the database |
| 07 | checkDuplicate() | The function check duplicate email when creating a new account |
| 08 | List<Account> searchName | The function used to query data,get account through keywords from the database |
| 09 | List<Account> getAllowPagination | The function used to pagination AccountPage |

AccountController

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doGet() GetAccountDetail | The function displays all account details |
| 02 | doGet() ModifyAccountl | Function used to display user detail so that that can be edited |
| 03 | doGet() RemoveAccount | Function used to ban an account |
| 04 | doGet() SearchAccountByName | Function used to find an account by their username |
| 05 | doGet() SortAccountByName | Function used to sort an account by name |
| 06 | doGet() SearchAccountByRole | Function used to find an account by their role |

### c. Sequence Diagram(s)

****

### d. Database Queries

– select \* from tblAccount where account\_id=?

– select \* from tblAccount

– update tblAccount set account\_is\_active =? where account\_id=?

– select account\_email from tblAccount where account\_email =?

– select \* from tblAccount where account\_name Like ?

– SELECT DATEPART(MONTH, create\_date) AS [Month], COUNT(account\_id) AS [total] FROM tblAccount where role\_id=? and YEAR(create\_date)=? GROUP BY DATEPART(MONTH, [create\_date]) ORDER BY [Month]

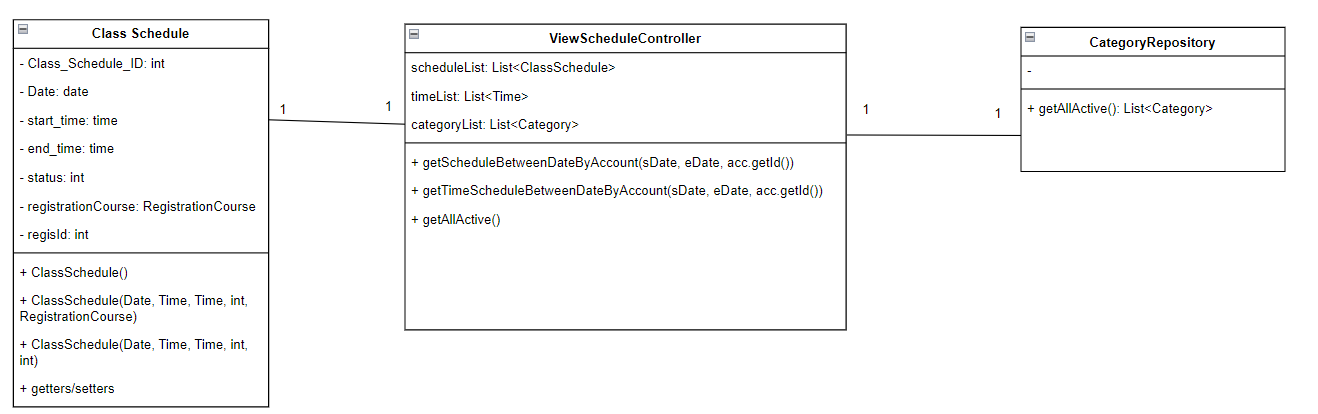
– select YEAR(create\_date) as year from tblAccount where role\_id=? group by YEAR(create\_date) order by YEAR(create\_date) desc

– select \* from tblAccount order by account\_id desc OFFSET ? ROWS FETCH NEXT? ROWS ONLY

– select count(\*) as num from tblAccount

## 4. View Timetable

### a. Class Diagram



### b. Class Specifications

#### Class Schedule Model

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter/setter | Get value and set value for attributes |
| 02 | Constructor | Generate Class Schedule object |

#### 

#### ClassScheduleRepository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getScheduleBetweenDateByAccount(Date startDate,Date endDate, int accountId) | Function used to query data, get list date in the week by account ID from the database |
| 02 | getTimeScheduleBetweenDateByAccount(Date startDate, Date endDate, int accountId) | Function used to query data, get list time in date by account ID from the database |

#### 

#### CategoryRepository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAllActive() | Function used to query data, get category list has category\_is\_active = 1 from database |

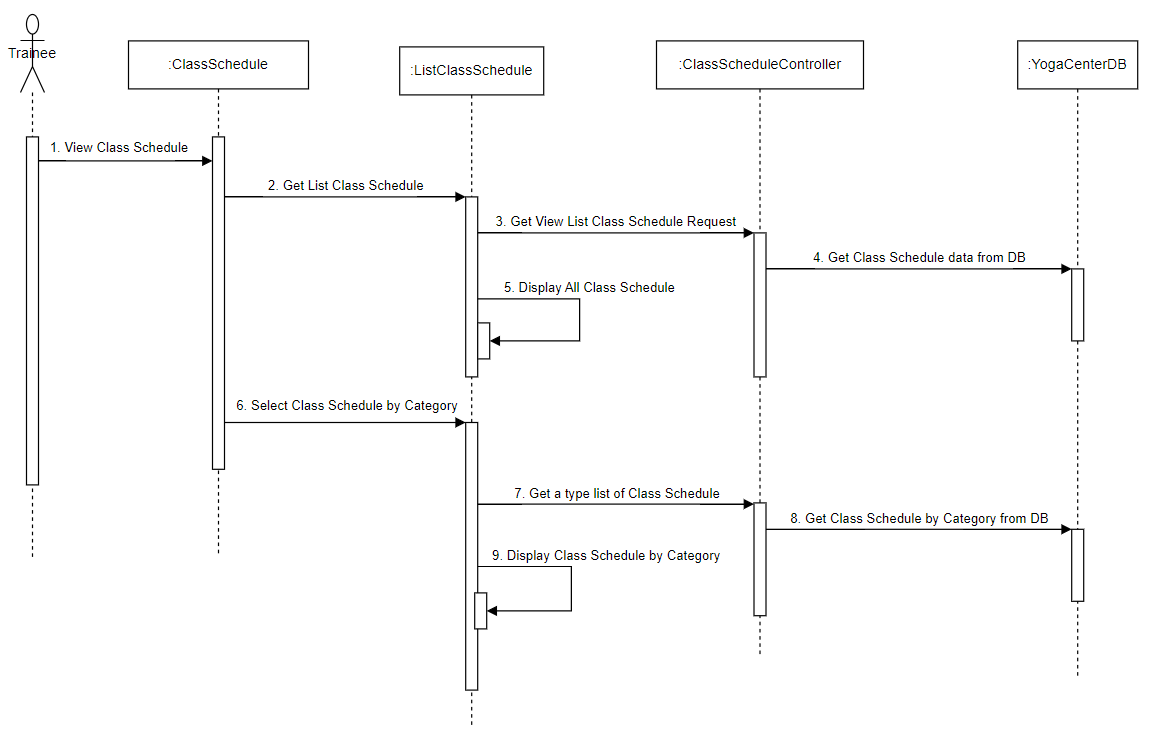
#### 

#### ViewScheduleController Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doget() | Function to get current week in timetable |
| 02 | doPost() | Function to formatting date |
| 03 | -loadScheduleOfDate(LocalDate date, HttpServletRequest request) | Function to load data for a date in a week |

#### 

### c. Sequence Diagram(s)

***d. Database Queries***

***SELECT*** *cs.slot\_start\_time*

***FROM***

*(****SELECT*** *slot\_start\_time, registration\_id*

***FROM*** *tblClassSchedule*

***WHERE*** *class\_date* ***BETWEEN***  *[slot\_start\_time]* ***AND*** *[slot\_end\_time]) cs*

***JOIN***

*(****SELECT*** *registration\_id*

***FROM*** *tblRegistrationCourse*

***WHERE*** *account\_id=[accountId] ) rc*

***ON*** *cs.registration\_id=rc.registration\_id*

***GROUP BY*** *slot\_start\_time*

***ORDER BY*** *slot\_start\_time* ***ASC***

***SELECT*** *\**

***FROM***

*(****SELECT*** *\**

***FROM*** *tblClassSchedule*

***WHERE*** *class\_date BETWEEN [slot\_start\_time] AND [slot\_end\_time]) cs*

***JOIN***

*(****SELECT*** *registration\_id*

***FROM*** *tblRegistrationCourse*

***WHERE*** *account\_id=[accountId] AND registration\_status=1 ) rc*

***ON*** *cs.registration\_id=rc.registration\_id*

***ORDER BY*** *class\_date* ***ASC*** *, slot\_start\_time* ***ASC***

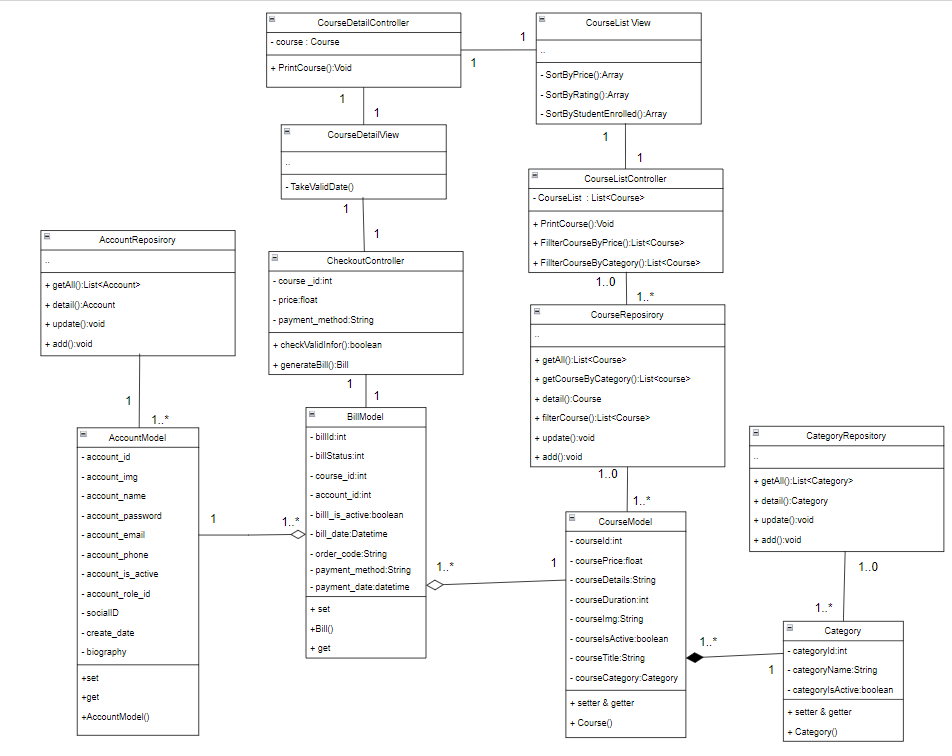
***SELECT*** *\**

***FROM*** *tblCategory*

***WHERE*** *category\_is\_active=1";*

## 5. Booking course

### a. Class Diagram



### b. Class Specifications

### AccountRepository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll() | Get all accounts in the database where the active field is true, there is no input parameter, the output is the account list |
| 02 | detail() | Get an account in the database where account\_id is equal to the input parameter, the output is an account |
| 03 | update() | Update an account in the database where account\_id is equal to the input parameter, with no output |
| 04 | add() | Add an account into the database, email is not duplicate, input is an account, no output |

#### 

***AccountModel Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | setter() | Set data for attributes in class |
| 02 | getter() | Get data value from attribute in class |
| 03 | AccountModel() | Constructor for class |

***CheckoutController Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | checkValidInfor() | Method check payment information from user |
| 02 | generateBill() | Method to generate bill base on payment information, output is bill  , input are course\_id, account\_id, price, Payment\_method, Payment\_date, order\_code, bill\_status |

***CategoryRepository Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll() | Get all category in database where active is true, output is category list, no input |
| 02 | detail() | Get a category in database where category\_id equal with input parameter, output is a category |
| 03 | update() | Update a category in database where category\_id equal with input category object, no output |
| 04 | add() | Add new category into database , input is a category object, no output |

***Category Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | setter() | Set data for attributes in class |
| 02 | getter() | Get data value from attribute in class |
| 03 | Category() | Constructor for class |

***CourseModel Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | setter() | Set data for attributes in class |
| 02 | getter() | Get data value from attribute in class |
| 03 | Course() | Constructor for class |

***CourseRepository Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll() | Get all course in database where active is true, output is course list, no input |
| 02 | getCourseByCategory() | Get all course in the database where category\_id equal with input parameter, output is course list |
| 03 | update() | Update a course in database where course\_id equal with input course object, no output |
| 04 | add() | Add new course into database , input is a course object, no output |
| 05 | filterCourse() | Get all course in database where course is satisfy: price in range min max, category\_id equal with input parameter, input include min , max price and category\_id, output is a course list |
| 06 | detail() | Get a course in database where course\_id equal with input parameter, output is an account |

***CourseListController Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | PrintCourse() | Give course list an print at CourseListview page |
| 02 | FilterCourseByPrice() | Take data filter by using CourseRepositoy, input is price min, price max, output is course list |
| 03 | FilterCourseByCategory() | Take data filter by using courseRepository, input is category, output is course list |

***CourseListView Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | SortByPrice() | Sort array course follow price by using build in method sort(), input is array course, output is array sorted course |
| 02 | SortByRating() | Sort array course follow rating star by using build in method sort(), input is array course, output is array sorted course |
| 03 | SortByStudentEnrolled() | Sort array course follow rating star by using build in method sort(), input is array course, output is array sorted course |

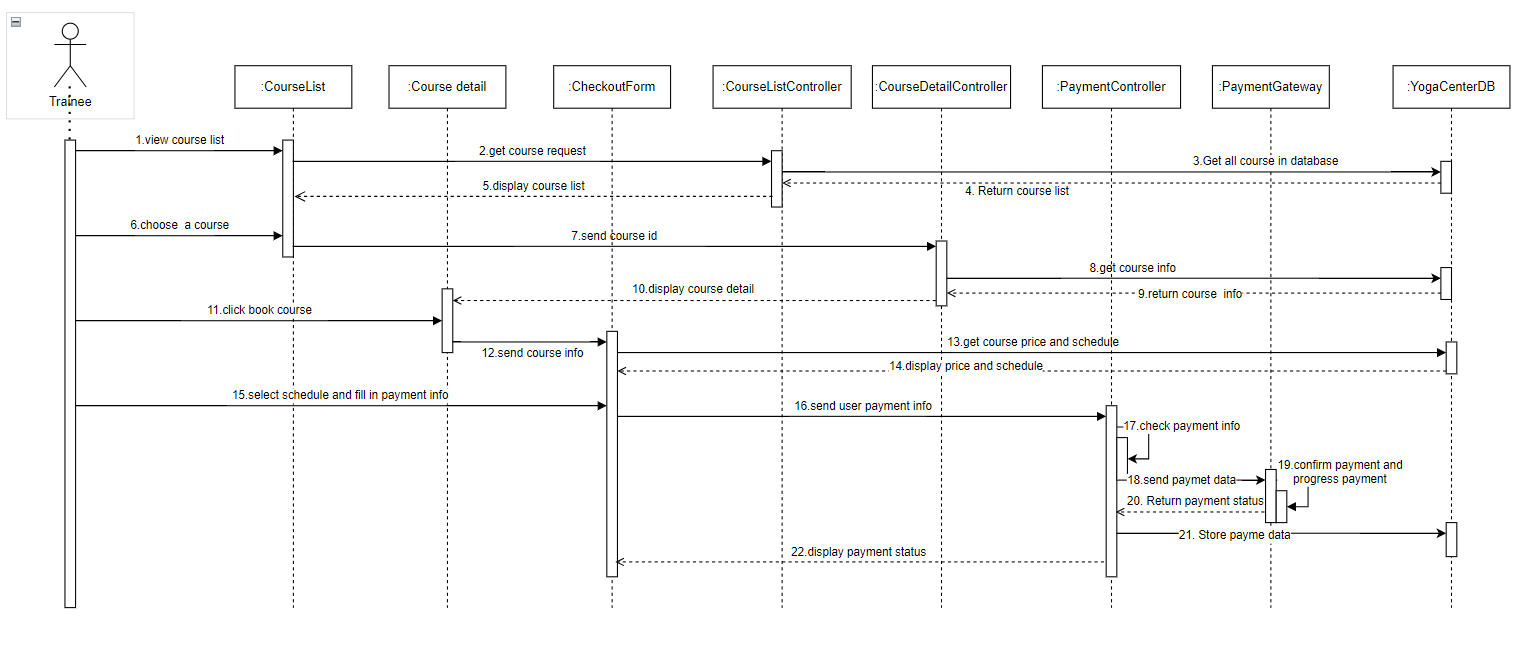
***CourseDetailController Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | PrintCourse() | Print course info at courseDetailView |

***CourseDetailView Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | TakeValidDate() | check valid date from user when booking course, no in put, output is date |

### c. Sequence Diagram(s)



### d. Database Queries

*SELECT \* FROM tblCategory WHERE category\_is\_active = 1;*

*SELECT \* FROM tblCategory WHERE category\_id = <category\_id>;*

*UPDATE tblCategory SET category\_name = <category\_name>, category\_is\_active = <category\_is\_active> WHERE category\_id = <category\_id>;*

*INSERT INTO tblCategory (category\_name, category\_is\_active) VALUES (<category\_name>, <category\_is\_active>);*

*UPDATE tblCourse SET course\_title = <course\_title>, course\_is\_active = <course\_is\_active>, course\_detail = <course\_detail>, course\_duration = <course\_duration>, course\_img = <course\_img>, course\_price = <course\_price>, category\_id = <category\_id>, account\_id = <account\_id> WHERE course\_id = <course\_id>;*

*SELECT \* FROM tblCourse WHERE course\_id = <course\_id>;*

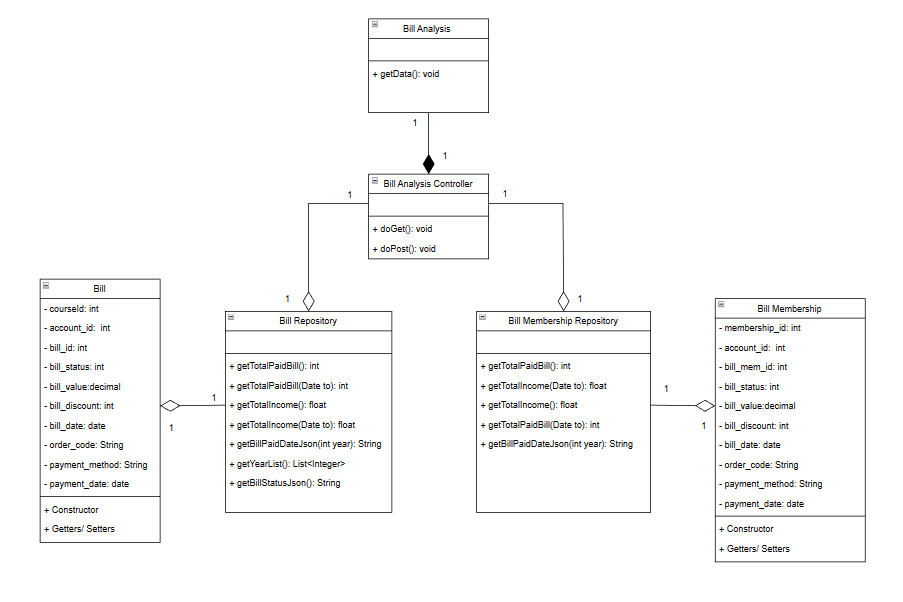
*SELECT \* FROM tblCourse WHERE course\_is\_active = 1 AND category\_id = <category\_id>;*

*SELECT \* FROM tblCourse WHERE course\_is\_active = 1;*

*INSERT INTO tblBill (course\_id, account\_id, bill\_status, bill\_is\_active, bill\_value, bill\_discount, bill\_date, order\_code, payment\_method) VALUES (<course\_id>, <account\_id>, <bill\_status>, <bill\_is\_active>, <bill\_value>, <bill\_discount>, <bill\_date>, <order\_code>, <payment\_method>)*

## 6. Bill Analysis

### a. Class Diagram



### b. Class Specifications

#### Bill Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter | Get value in attributes |
| 02 | Setter | Set value for attributes |
| 03 | Constructor | Generate Bill object |

#### 

#### Bill Membership Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter | Get value in attributes |
| 02 | Setter | Set value for attributes |
| 03 | Constructor | Generate Bill Mebership object |

#### Bill Repository

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getTotalPaidBill() | Get numbers of bills that have the status “0” |
| 02 | getTotalPaidBill(Date to) | Get numbers of bills that have the status “0” and have payment\_date <= “to” |
| 03 | getTotalIncome() | Get the total income of all bills |
| 04 | getTotalIncome(Date to) | Get the total income of all bills that have payment\_date <= “to” |
| 05 | getBillPaidDateJson(int year) | Get the total number of bills for each month |
| 06 | getYearList() | Get list year of the payment\_date |
| 07 | getBillStatusJson() | Get the total number of bills for each status |

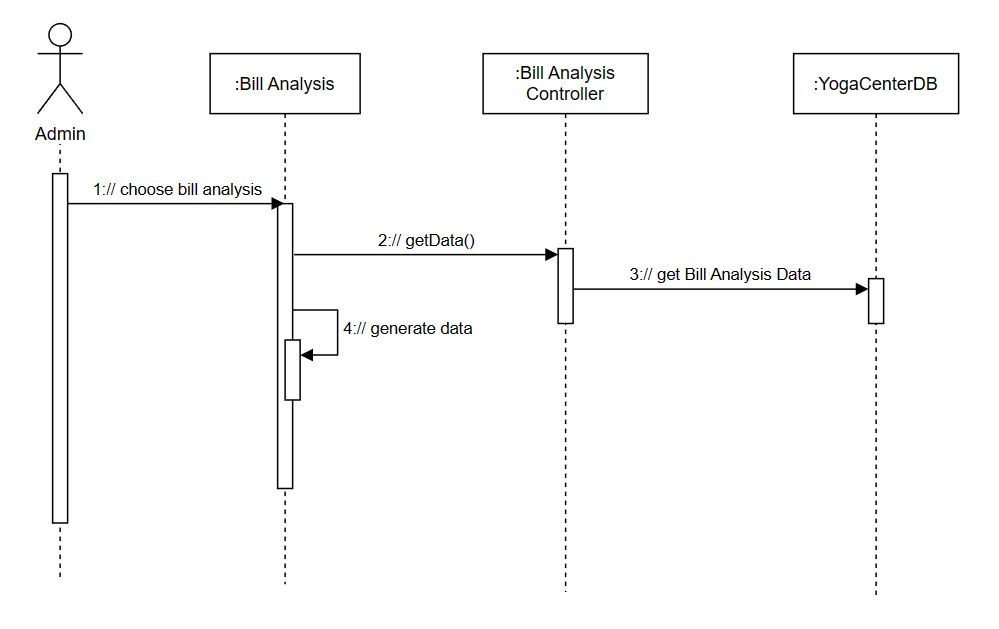
#### Bill Membership Repository

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getTotalPaidBill() | Get numbers of bills that have the status “0” |
| 02 | getTotalPaidBill(Date to) | Get numbers of bills that have the status “0” and have payment\_date <= “to” |
| 03 | getTotalIncome() | Get the total income of all bills |
| 04 | getTotalIncome(Date to) | Get the total income of all bills that have payment\_date <= “to” |
| 05 | getBillPaidDateJson(int year) | Get the total number of bills for each month |

#### Bill Analysis Controller

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doGet() | Get data from Yoga Center Database, generate and set attributes into request than forward to ViewBillAnalysis.jsp |
| 02 | doPost() | Input is select tag in ViewBillAnalysis.jsp. Get input then generate chart follow year. |

### c. Sequence Diagram(s)



### d. Database Queries

SELECT count(\*) as num FROM tblBill WHERE bill\_status = 0

SELECT count(\*) as num FROM tblBill WHERE bill\_status = 0 and payment\_date<=?

SELECT SUM(bill\_value) as income FROM tblBill WHERE bill\_status = 0

SELECT SUM(bill\_value) as income FROM tblBill WHERE bill\_status = 0 and payment\_date<=?

SELECT DATEPART(MONTH, payment\_date) AS [Month], COUNT(bill\_id) AS [total] FROM tblBill WHERE bill\_status=0 and YEAR(payment\_date)=? GROUP BY DATEPART(MONTH, [payment\_date]) ORDER BY [Month]

SELECT YEAR(payment\_date) as year FROM tblBill WHERE bill\_status = 0 GROUP BY YEAR(payment\_date) ORDER BY YEAR(payment\_date) desc

SELECT bill\_status AS status, COUNT(\*) AS total\_count FROM ( SELECT bill\_status FROM tblBill UNION ALL SELECT bill\_status FROM tblBillMembership ) AS combined\_tables GROUP BY bill\_status

SELECT count(\*) as num FROM tblBillMembership WHERE bill\_status = 0

SELECT count(\*) as num FROM tblBillMembership WHERE bill\_status = 0 and payment\_date<=?

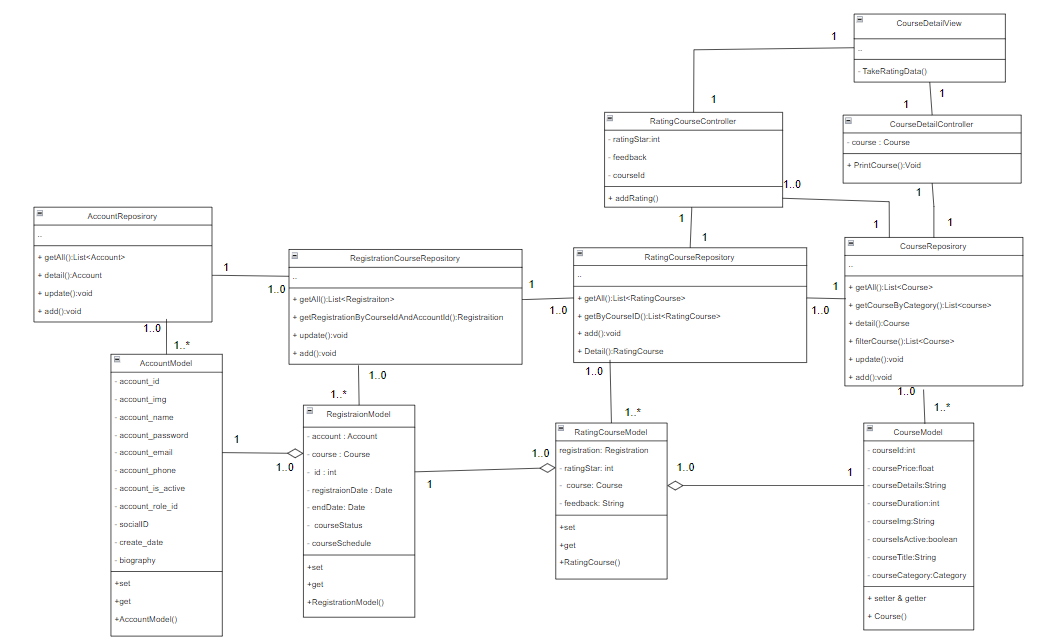
SELECT SUM(bill\_value) as income FROM tblBillMembership WHERE bill\_status = 0

SELECT SUM(bill\_value) as income FROM tblBillMembership WHERE bill\_status = 0 and payment\_date<=?

SELECT DATEPART(MONTH, payment\_date) AS [Month], COUNT(bill\_mem\_id) AS [total] FROM tblBillMembership where bill\_status=0 and YEAR(payment\_date)=? GROUP BY DATEPART(MONTH, [payment\_date]) ORDER BY [Month]

## 7. Rating course

### a. Class Diagram



### b. Class Specifications

***CourseModel Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | setter() | Set data for attributes in class |
| 02 | getter() | Get data value from attribute in class |
| 03 | Course() | Constructor for class |

***CourseRepository Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll() | Get all course in database where active is true, output is course list, no input |
| 02 | getCourseByCategory() | Get all course in database where category\_id equal with input parameter, output is course list |
| 03 | update() | Update a course in database where course\_id equal with input course object, no output |
| 04 | add() | Add new course into database , input is a course object, no output |
| 05 | filterCourse() | Get all course in database where course is satisfy: price in range min max, category\_id equal with input parameter, input include min , max price and category\_id, output is a course list |
| 06 | detail() | Get a course in database where course\_id equal with input parameter, output is an account |

### AccountRepository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll() | Get all accounts in the database where the active field is true, there is no input parameter, the output is the account list |
| 02 | detail() | Get an account in the database where account\_id is equal to the input parameter, the output is an account |
| 03 | update() | Update an account in the database where account\_id is equal to the input parameter, with no output |
| 04 | add() | Add an account into the database, email is not duplicate, input is an account, no output |

#### 

***AccountModel Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | setter() | Set data for attributes in class |
| 02 | getter() | Get data value from attribute in class |
| 03 | AccountModel() | Constructor for class |

***RatingCourseModel Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | setter() | Set data for attributes in class |
| 02 | getter() | Get data value from attribute in class |
| 03 | RatingCourseModel() | Constructor for class |

***RegistrationCourseRepository Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | setter() | Set data for attributes in class |
| 02 | getter() | Get data value from attribute in class |

***RatingCourseRepository Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll() | Get all rating course in database, no input , output is a list rating course |
| 02 | getByCourseId() | Get all rating course in database where course id equal with course id input parameter, output is a list rating course |
| 03 | Add() | Add new a rating course to database where each rating have different registration course id different, input is a rating course object data, no output |
| 04 | Detail() | Get an rating course in database where course\_id and registration id equal with input parameter, output is a rating course |

***RatingCourseController Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | addRating() | Method take info of from request parameter and inster to database by using RatingCourseRepository |

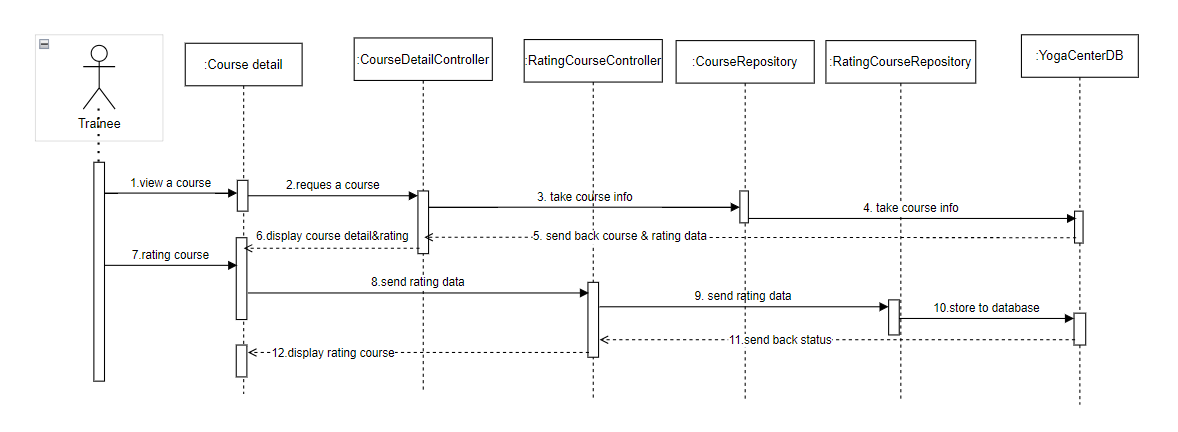
***CourseDetailController Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | PrintCourse() | Print course info at courseDetailView |

***CourseDetailView Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | TakeRatingData() | create ui, take data and send rating data to CourseRatingController when user click button “post”, no input and output. |

### c. Sequence Diagram(s)



### d. Database Queries

*SELECT \* FROM tblRatingCourse*

*SELECT \* FROM tblRatingCourse WHERE course\_id = <course\_id>*

*SELECT \* FROM tblRatingCourse WHERE registration\_id = <registration\_id>*

*INSERT INTO tblRatingCourse (registration\_id, feedback, rating\_star, course\_id) VALUES (<registration\_id>, <feedback>, <rating\_star>, <course\_id>)*

*UPDATE tblCourse SET course\_title = <course\_title>, course\_is\_active = <course\_is\_active>, course\_detail = <course\_detail>, course\_duration = <course\_duration>, course\_img = <course\_img>, course\_price = <course\_price>, category\_id = <category\_id>, account\_id = <account\_id> WHERE course\_id = <course\_id>;*

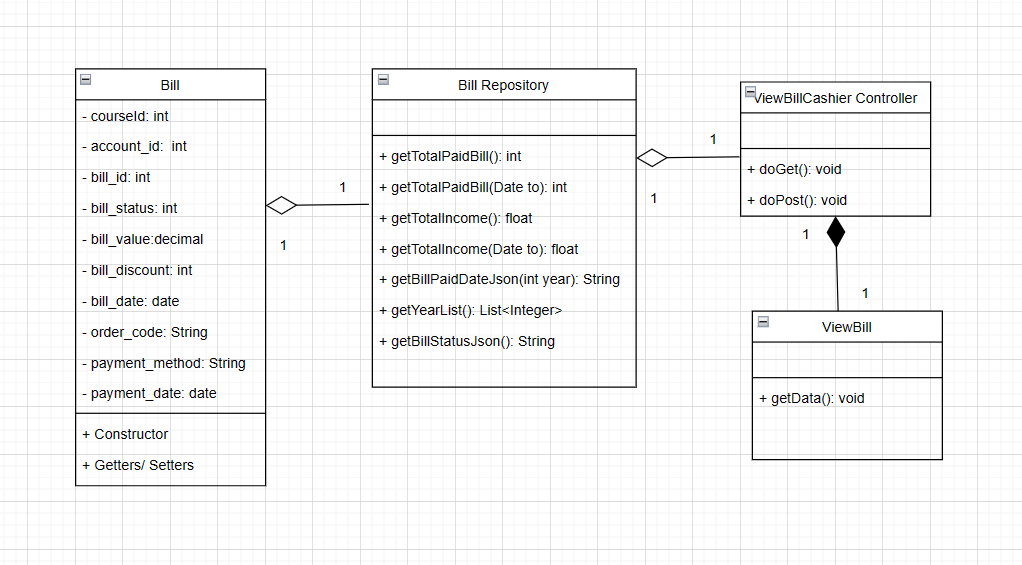
*SELECT \* FROM tblCourse WHERE course\_id = <course\_id>;*

*SELECT \* FROM tblCourse WHERE course\_is\_active = 1 AND category\_id = <category\_id>;*

*SELECT \* FROM tblCourse WHERE course\_is\_active = 1;*

## 8. Cashier ViewBill

### a. Class Diagram

****

### b. Class Specifications

***BillModel Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | setter() | Set data for attributes in class |
| 02 | getter() | Get data value from attribute in class |
| 03 | Bill() | Constructor for class |
| 04 | BillStatus{COMPLETED,CANCELLED,PENDING} | Three statuses of Bill and it will change when the cashier updates the status |
| 05 | BillStatus getEnumIndex(int i) | Get each element in the array |

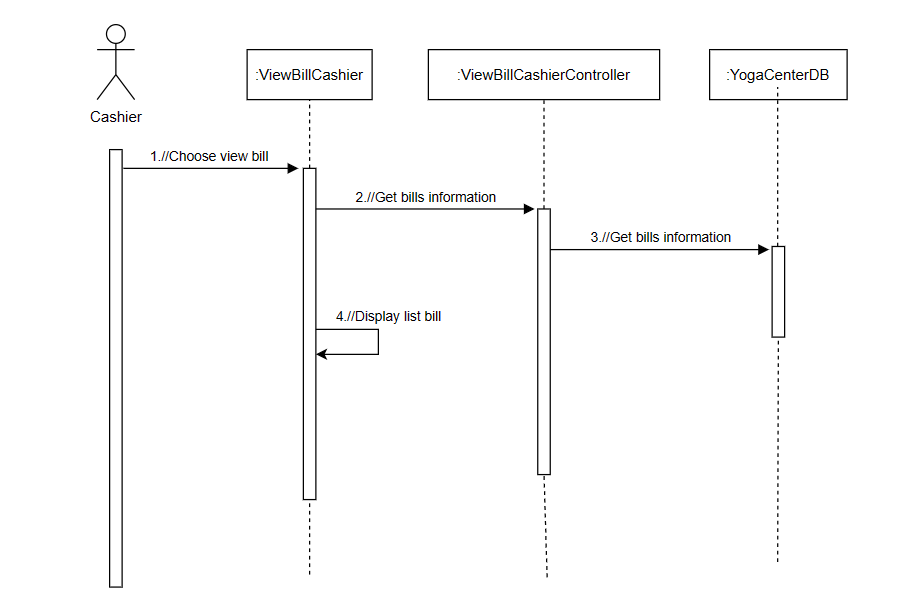
***BillRepository Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll() | Get all bill in database, no input, output is a list of bill |
| 02 | getCourseIdByOrdercode(String ordercode) | Get all bill where order\_code = ? |
| 03 | getAllByAccountIdAndCourseID(String accountId, int courseId ) | Get all bill where account\_id = ? AND course\_id = ? |
| 04 | updateStatus(String ordercode, LocalDate date, int status) | Update bill\_status and payment\_date where order\_code = ? |
| 05 | detail(int id) | Get all bill where bill\_id = ? |

***ViewBillCashierController Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doGet() | Method get all bill by using BillRepository |

### c. Sequence Diagram(s)

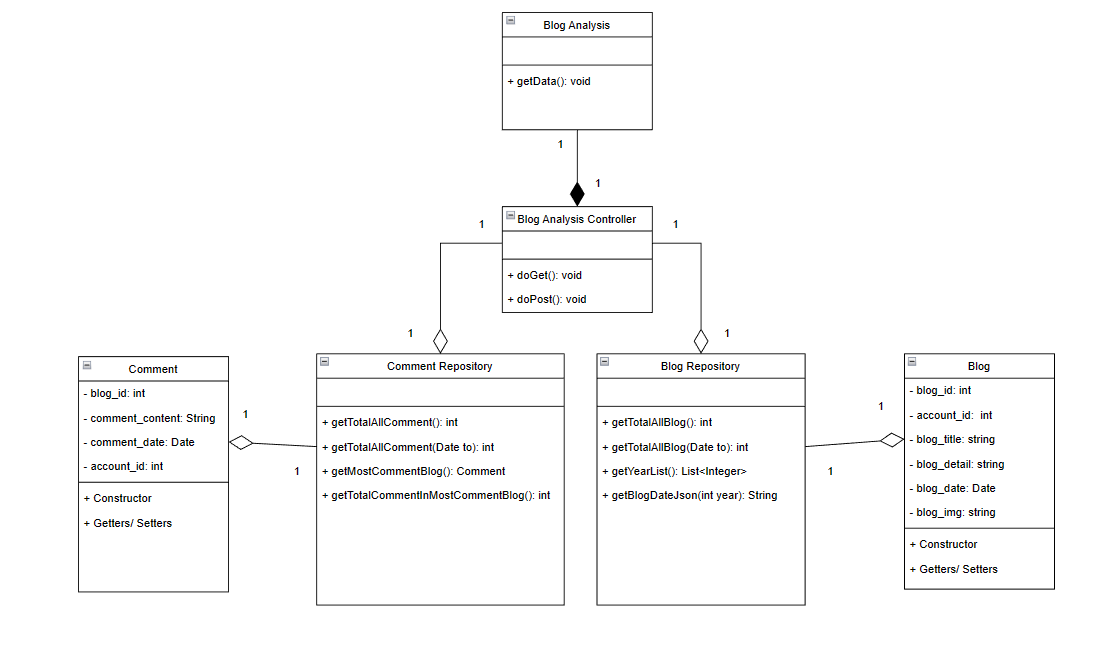


### d. Database Queries

select \* from tblBill

## 9. Blog Analysis

### a. Class Diagram



### b. Class Specifications

#### Blog Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter | Get value in attributes |
| 02 | Setter | Set value for attributes |
| 03 | Constructor | Generate Blog object |

#### 

#### Comment Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter | Get value in attributes |
| 02 | Setter | Set value for attributes |
| 03 | Constructor | Generate Comment object |

#### Blog Repository

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getTotalAllBlog() | Get numbers of blog that have the status “0” |
| 02 | getTotalAllBlog(Date to) | Get numbers of blog that have the status “0” and have blog\_date <= “to” |
| 03 | getBlogDateJson(int year) | Get the total number of blog for each month |
| 04 | getYearList() | Get list year of the blog\_date |

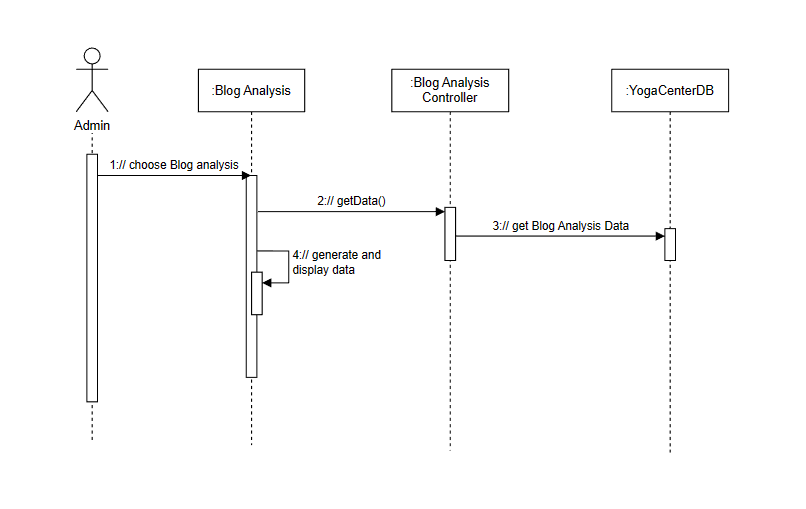
#### Comment Repository

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getTotalAllComment() | Get numbers of comment |
| 02 | getTotalAllComment(Date to) | Get numbers of comment that have comment\_date <= “to” |
| 03 | getMostCommentBlog() | Get the blog that have most comments |
| 04 | getTotalCommentInMostCommentBlog() | Get the total comments in the blog that have most comments |

#### Blog Analysis Controller

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doGet() | Get data from Yoga Center Database, generate and set attributes into request than forward to ViewBlogAnalysis.jsp |
| 02 | doPost() | Input is select tag in ViewBlogAnalysis.jsp. Get input then generate chart follow year. |

### c. Sequence Diagram(s)



### d. Database Queries

SELECT count(\*) as num FROM tblBlog

SELECT count(\*) as num FROM tblBlog WHERE blog\_date <= ?

SELECT DATEPART(MONTH, blog\_date) AS [Month], COUNT(blog\_id) AS [total] FROM tblBlog WHERE YEAR(blog\_date)=? GROUP BY DATEPART(MONTH, [blog\_date]) ORDER BY [Month]

SELECT YEAR(blog\_date) as year FROM tblBlog GROUP BY YEAR(blog\_date) order by YEAR(blog\_date) DESC

SELECT count(\*) as num from tblComment

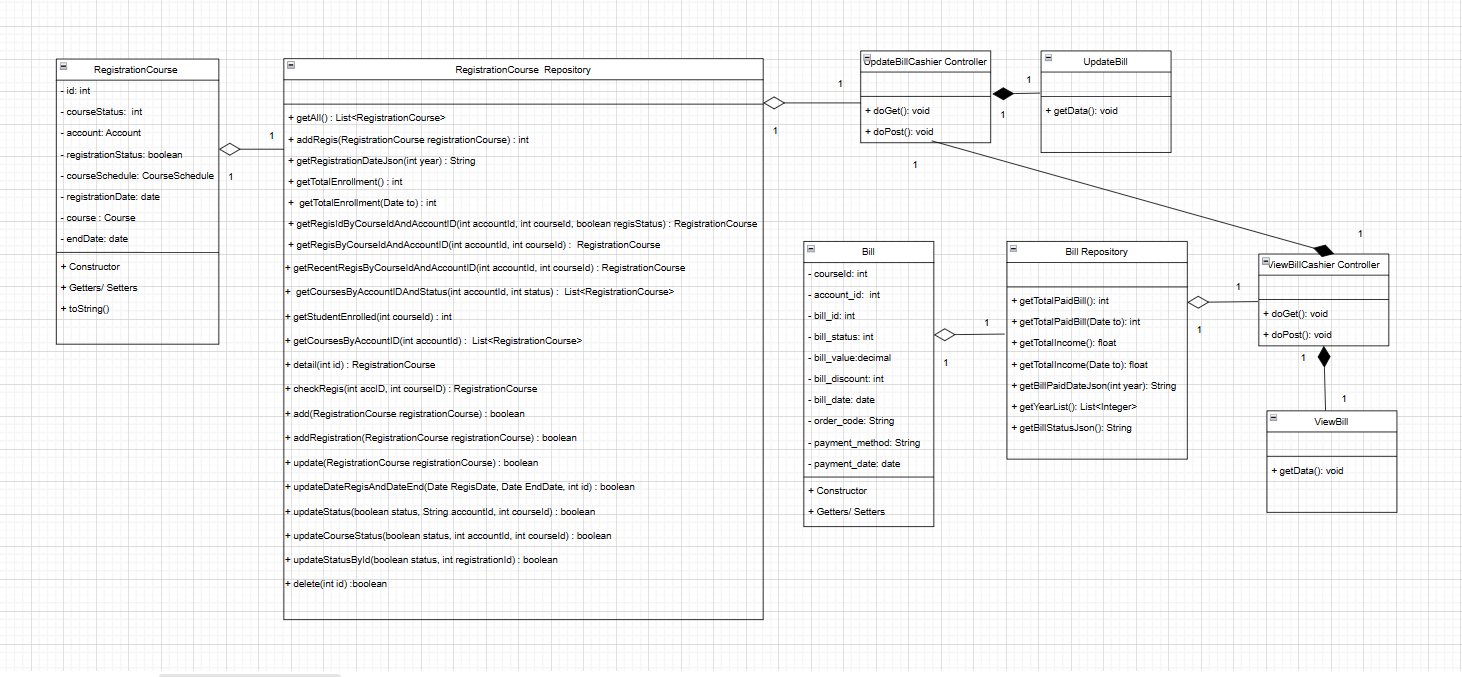
SELECT count(\*) as num from tblComment where comment\_date <= ?

SELECT blog\_id , COUNT(comment\_content) as total FROM tblComment GROUP BY blog\_id ORDER BY total DESC OFFSET 0 ROWS FETCH NEXT 1 ROWS ONLY

SELECT blog\_id , COUNT(comment\_content) as total FROM tblComment GROUP BY blog\_id ORDER BY total DESC OFFSET 0 ROWS FETCH NEXT 1 ROWS ONLY

## 10. Cashier UpdateBill

### a. Class Diagram



### b. Class Specifications

***BillModel Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | setter() | Set data for attributes in class |
| 02 | getter() | Get data value from attribute in class |
| 03 | Bill() | Constructor for class |
| 04 | BillStatus{COMPLETED,CANCELLED,PENDING} | Three statuses of Bill and it will change when the cashier updates the status |
| 05 | BillStatus getEnumIndex(int i) | Get each element in the array |

***BillRepository Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll() | Get all bill in database, no input, output is a list of bill |
| 02 | getCourseIdByOrdercode(String ordercode) | Get all bill where order\_code = ? |
| 03 | getAllByAccountIdAndCourseID(String accountId, int courseId ) | Get all bill where account\_id = ? AND course\_id = ? |
| 04 | updateStatus(String ordercode, LocalDate date, int status) | Update bill\_status and payment\_date where order\_code = ? |
| 05 | detail(int id) | Get all bill where bill\_id = ? |

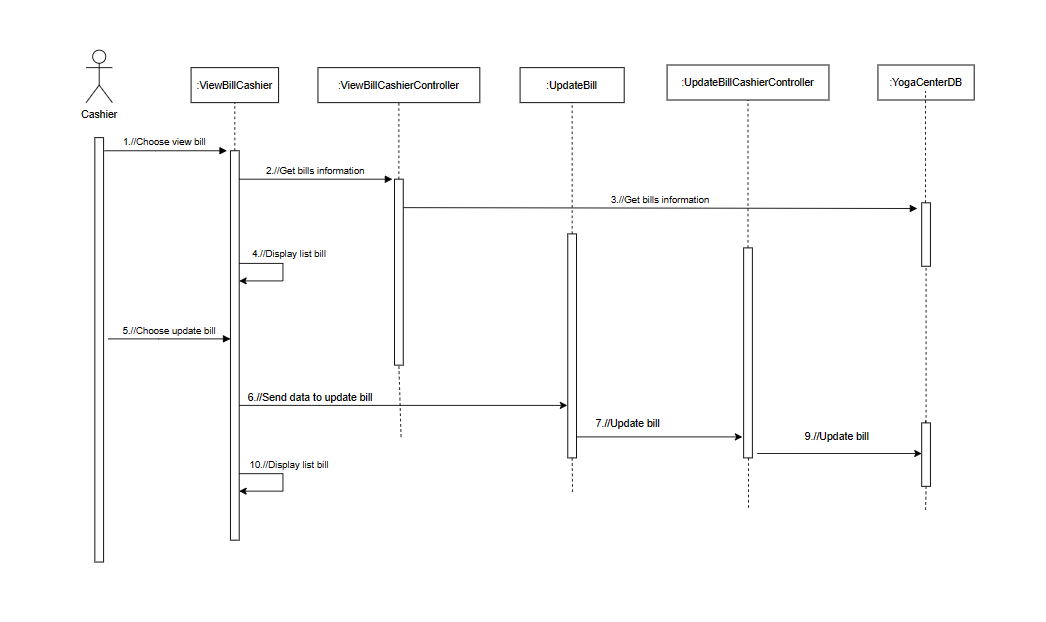
***RegistrationCourseRepository Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getRecentRegisByCourseIdAndAccountID(int accountId, int courseId) | Get all registration course where account\_id = ? AND course\_id = ? , and order by registration\_id desc |
| 02 | updateStatusById(boolean status, int registrationId) | Update registration\_status WHERE registration\_id = ? |

***UpdateBillCashierController Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doGet() | Get information details to update form |
| 02 | doPost() | Update bill\_status and payment\_date and update registration\_status by using BillRepository and RegistrationCourseRepository |

### c. Sequence Diagram(s)



### d. Database Queries

select \* from tblBill where bill\_id=?

UPDATE tblBill SET bill\_status = ? , payment\_date =? WHERE order\_code = ?

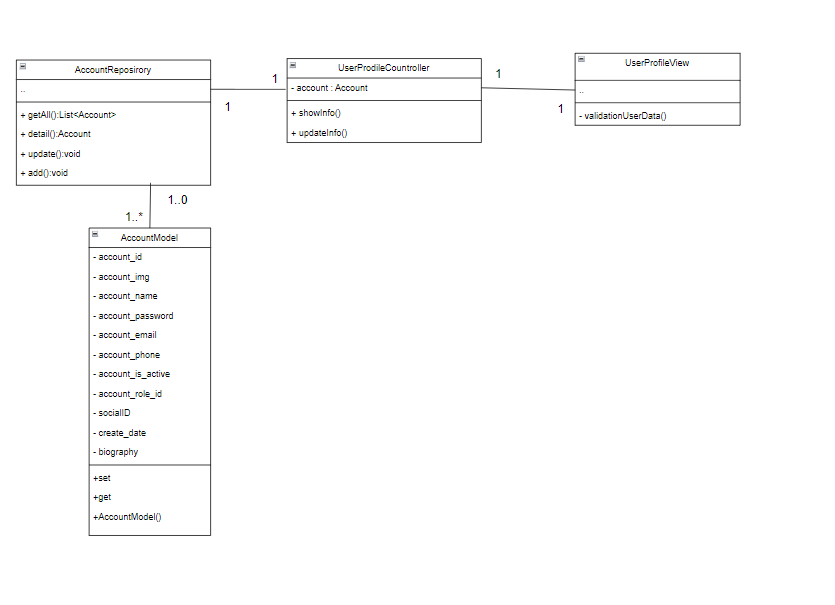
select \* from tblBill where order\_code=?

select \* from tblRegistrationCourse WHERE account\_id = ? AND course\_id = ? order by registration\_id desc

UPDATE tblRegistrationCourse SET registration\_status = ? WHERE registration\_id = ?

## 11. Update User Profile

### a. Class Diagram



### b. Class Specifications

### AccountRepository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll() | Get all accounts in the database where the active field is true, there is no input parameter, and the output is the account list |
| 02 | detail() | Get an account in the database where account\_id is equal to the input parameter, the output is an account |
| 03 | update() | Update an account in the database where account\_id is equal to the input parameter, with no output |
| 04 | add() | Add an account into the database, email is not duplicate, input is an account, no output |

#### 

***AccountModel Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | setter() | Set data for attributes in class |
| 02 | getter() | Get data value from attribute in class |
| 03 | AccountModel() | Constructor for class |

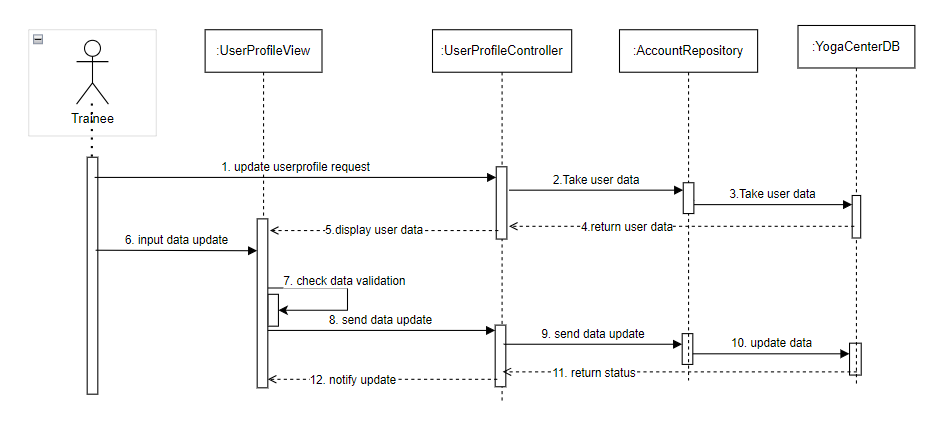
***UserProfileController Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | showInfo() | Take user data from database and show on UserProfileView, input is account id, no output |
| 02 | updateInfo() | Take info from UserProfileView and update to database, input is an account object, output is boolean |

***UserProfileView Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | validationData() | Method get user data input and check validation, input is name, phone, email, and password, output is boolean |

### c. Sequence Diagram(s)



### d. Database Queries

*INSERT INTO tblAccount ( account\_img, account\_name, account\_password, account\_email, account\_phone, account\_is\_active, role\_id, social\_id, biography) VALUES (<account\_img>, <account\_name>, <account\_password>, <account\_email>, <account\_phone>, <account\_is\_active>, <role\_id>, <social\_id>, <biography>)*

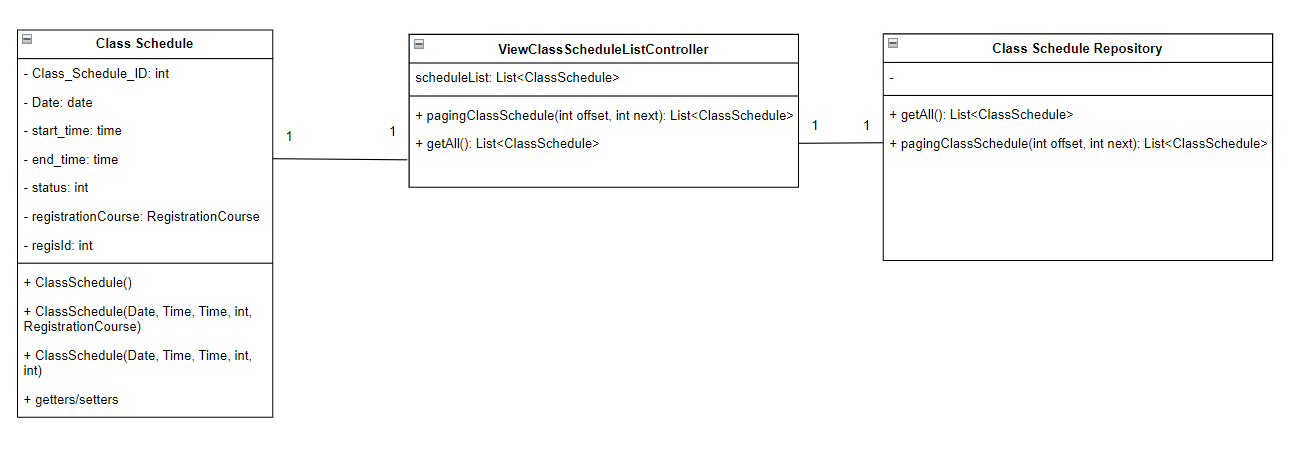
*UPDATE tblAccount SET account\_name=<account\_name> , account\_password= <account\_password> ,account\_img= <account\_img> , account\_email= <account\_email> , account\_phone= <account\_phone> , account\_is\_active= <account\_is\_active>, biography= <biography> where account\_id= <where account\_id>*

*SELECT \* FROM tblAccount where account\_id= <account\_id>*

*SELECT \* FROM tblAccount*

## 12. View Timetable List

### a. Class Diagram



### b. Class Specifications

***Class Schedule Model***

| No | Method | Description |
| --- | --- | --- |
| 01 | Getter/setter | Get value and set value for attributes |
| 02 | Constructor | Generate Class Schedule object |

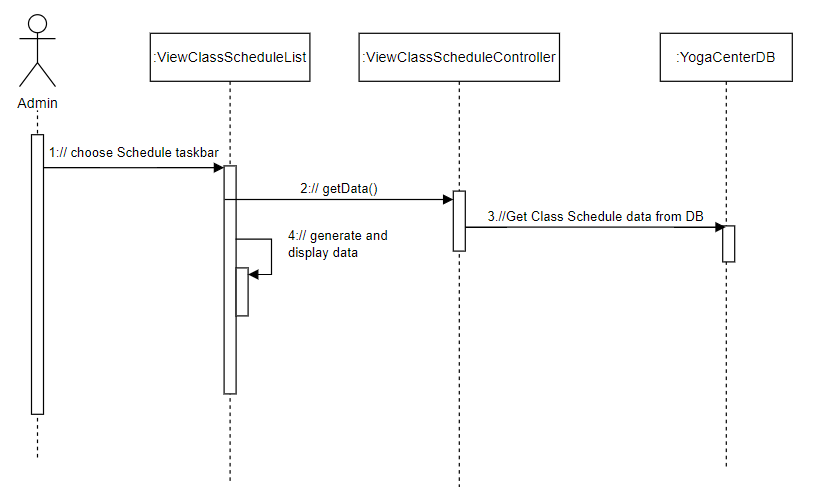
**ClassScheduleRepository Class**

| No | Method | Description |
| --- | --- | --- |
| 01 | pagingClassSchedule(int offset, int next) | Function used to get a list of data in the database and use OFFSET to paginate this data list |

**ViewClassScheduleListController Class**

| No | Method | Description |
| --- | --- | --- |
| 01 | doget() | Function to getting list data then request dispatch into JSP page |

### c. Sequence Diagram(s)



### d. Database Queries

SELECT \*

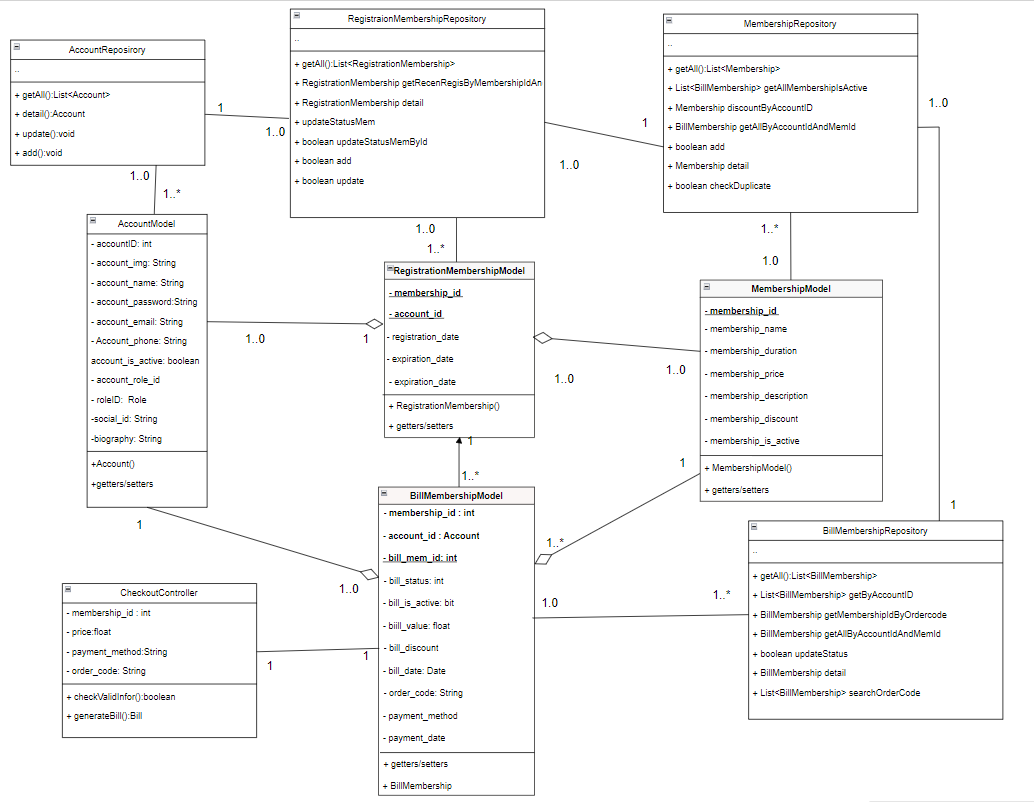
FROM tblClassSchedule

ORDER BY class\_schedule\_id

OFFSET ? ROWS FETCH NEXT ? ROWS ONLY

## 13. Buy membership

### a. Class Diagram



### b. Class Specifications

#### Membership Model Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter() | Get value in attributes |
| 02 | Setter() | Set value for attributes |
| 03 | Membership() | Generate Course object |

#### Account Model Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Account() | Constructor for class. |
| 02 | setter() | Method to set the date in class. |
| 03 | getter() | Method to get the date from class. |

#### BillMembership Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter() | Get value in attributes |
| 02 | Setter() | Set value for attributes |
| 03 | BillMembership () | Generate Membership object |

#### RegistrationMembership Model Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | DisplayBill() | Use jsp, jstl, css to generate view for checkout |

#### PaymentGateway Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | vnpay\_payment(.) | Send information of the bill to payment gateway and get back attribute to store in database |

#### 

#### RegistrationMembership Repository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll():List<RegistrationMembership> | Get all registration membership in the database where active is true, the output is a registration membership list, no input |
| 02 | RegistrationMembershipDetail | Get all registration detail in the database by registration id =? |
| 03 | updateStatusMemById | Update status of registration\_status |
| 04 | add | Add new registration membership |

#### 

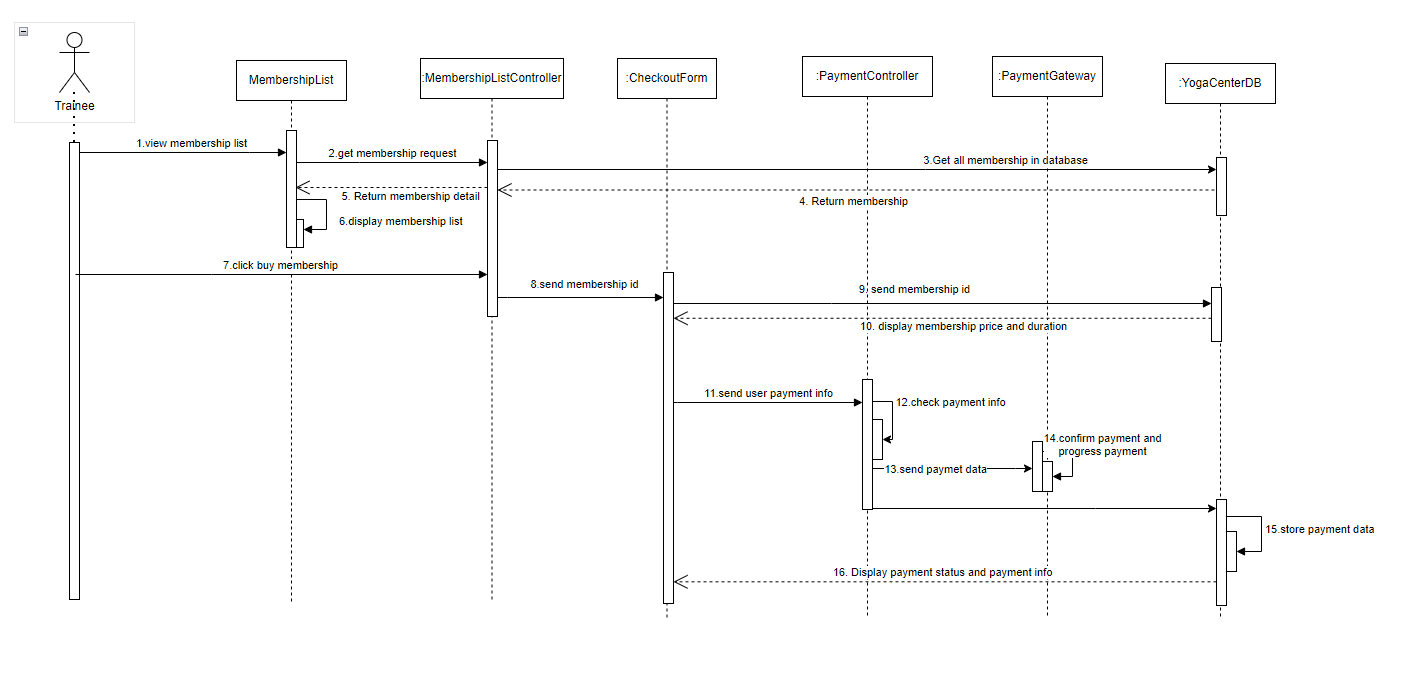
#### Membership Repository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll():List<Membership> | Get all membership in the database where active is true, the output is a membership list, no input |
| 02 | getAllByAccountIdAndMemId | Get membership current of an account |
| 03 | add | Add new membership |
| 04 | detail | Get the detail of membership |

***Account Repository Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | List<Account> getAll() | The function used to query data, get an account list from the database |
| 02 | updateAccount() | The function used to query data, edit data of account into the database |

### c. Sequence Diagram(s)



### d. Database Queries

AccountRepository  
select \* from tblRegistrationMembership

select \* from tblRegistrationMembership WHERE membership\_id = ? AND account\_id = ? order by membership\_id desc

select \* from tblRegistrationMembership where account\_id=?

UPDATE tblRegistrationMembership SET registration\_status = ?WHERE account\_id = ? AND membership\_id = ?

INSERT INTO tblRegistrationMembership (membership\_id, account\_id, registration\_date, expriration\_date, registration\_status) VALUES (?, ?, ?, ?, ?)

**RegistrationMembership Repository Class**

select \* from tblRegistrationMembership

INSERT INTO tblRegistrationMembership (membership\_id, account\_id, registration\_date, expriration\_date, registration\_status) VALUES (?, ?, ?, ?, ?)

**Membership Repository Class**

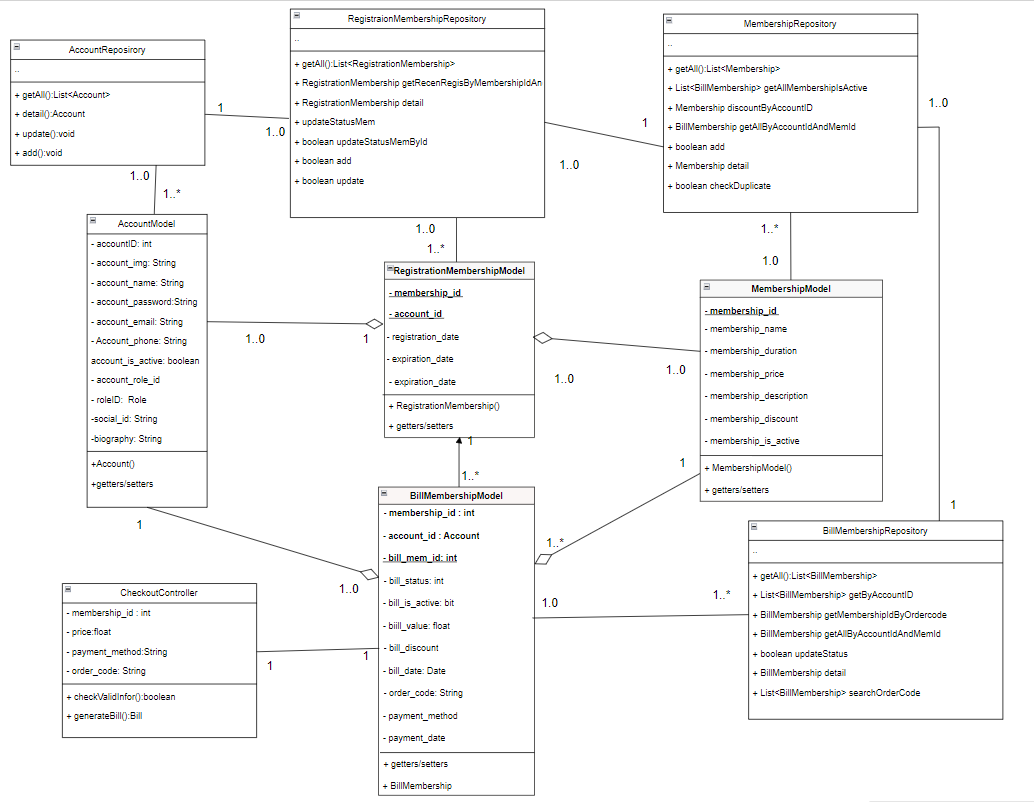
select \* from tblMembership

select \* from tblMembership where membership\_id=?

UPDATE tblMembership SET membership\_name = ?, membership\_duration = ?, membership\_price = ?, membership\_description = ?, membership\_discours = ?, membership\_is\_active = ? WHERE membership\_id = ?

## 14. View membership

### a. Class Diagram



### b. Class Specifications

#### Membership Model Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter() | Get value in attributes |
| 02 | Setter() | Set value for attributes |
| 03 | Membership() | Generate Course object |

#### Account Model Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Account() | Constructor for class. |
| 02 | setter() | Method to set the date in class. |
| 03 | getter() | Method to get the date from class. |

#### BillMembership Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter() | Get value in attributes |
| 02 | Setter() | Set value for attributes |
| 03 | BillMembership () | Generate Membership object |

#### RegistrationMembership Model Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | DisplayBill() | Use jsp, jstl, css to generate view for checkout |

#### PaymentGateway Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | vnpay\_payment(.) | Send information of the bill to payment gateway and get back attribute to store in database |

#### 

#### RegistrationMembership Repository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll():List<RegistrationMembership> | Get all registration membership in the database where active is true, the output is a registration membership list, no input |
| 02 | RegistrationMembershipDetail | Get all registration detail in the database by registration id =? |
| 03 | updateStatusMemById | Update status of registration\_status |
| 04 | add | Add new registration membership |

#### 

#### Membership Repository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll():List<Membership> | Get all membership in the database where active is true, the output is a membership list, no input |
| 02 | getAllByAccountIdAndMemId | Get membership current of an account |
| 03 | add | Add new membership |
| 04 | detail | Get the detail of membership |

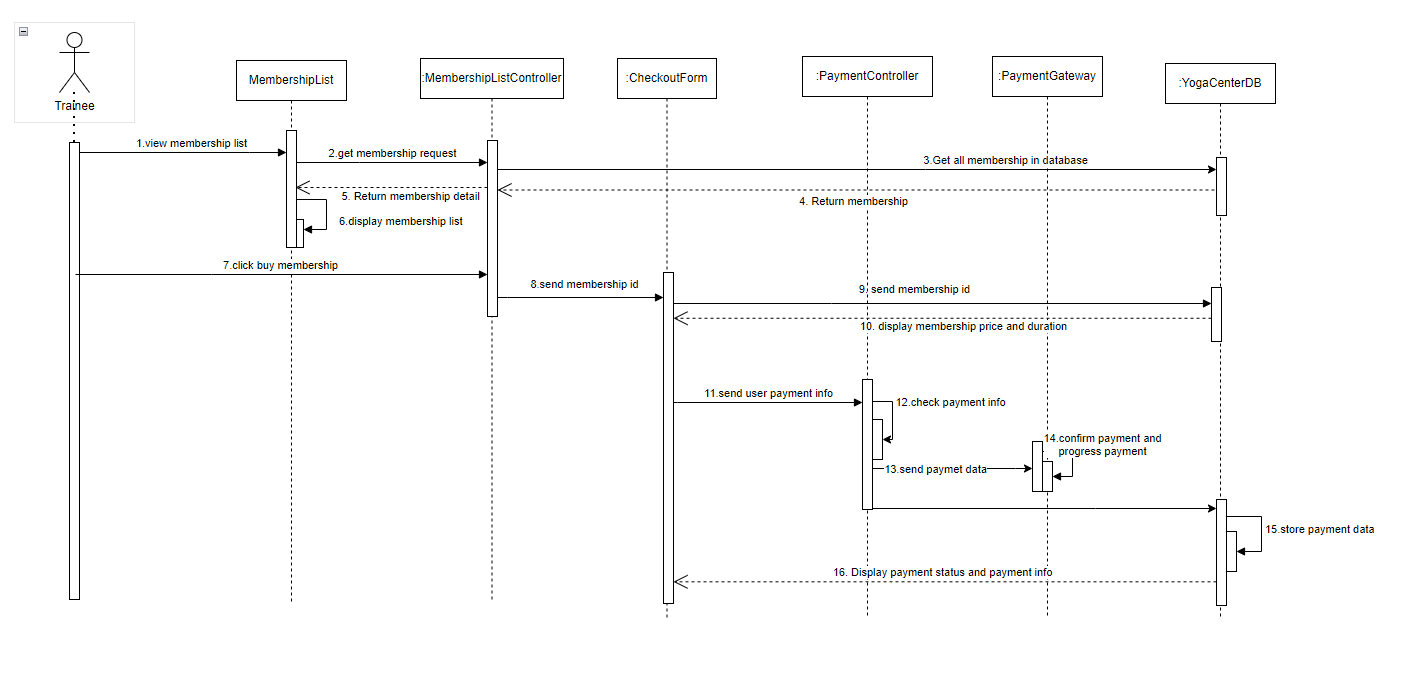
#### Bill Membership Repository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll():List<BillMembership> | Get all bill membership in the database where active is true, the output is a membership list, no input |
| 02 | getByAccountID | Get membership current of an account\_id |
| 03 | getMembershipIdByOrdercode | Get membership current of an order\_code |
| 04 | updateStatus | Update status of bill\_membership\_status where bill\_membership\_id=? |
| 05 | searchOrderCode | Search bill membership by order\_code=? |

***Account Repository Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | List<Account> getAll() | The function used to query data, get an account list from the database |
| 02 | updateAccount() | The function used to query data, edit data of account into the database |

### c. Sequence Diagram(s)



### d. Database Queries

**AccountRepository**  
select \* from tblRegistrationMembership

select \* from tblRegistrationMembership WHERE membership\_id = ? AND account\_id = ? order by membership\_id desc

INSERT INTO tblRegistrationMembership (membership\_id, account\_id, registration\_date, expriration\_date, registration\_status) VALUES (?, ?, ?, ?, ?)

**RegistrationMembership Repository Class**

select \* from tblRegistrationMembership

select \* from tblRegistrationMembership WHERE membership\_id = ? AND account\_id = ? order by membership\_id desc

select \* from tblRegistrationMembership where account\_id=?

UPDATE tblRegistrationMembership SET registration\_status =? WHERE account\_id = ? AND membership\_id = ?

INSERT INTO tblRegistrationMembership (membership\_id, account\_id, registration\_date, expriration\_date, registration\_status) VALUES (?, ?, ?, ?, ?)

**Membership Repository Class**

select \* from tblMembership

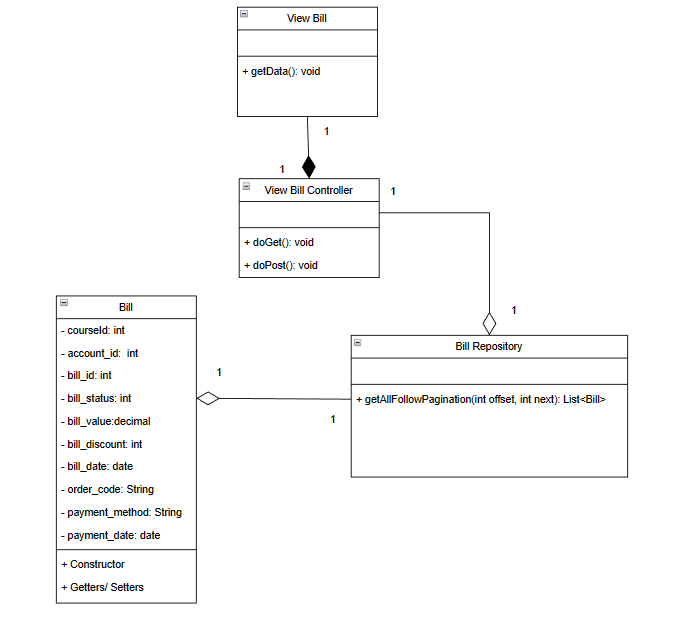
select \* from tblMembership where membership\_id=?

UPDATE tblMembership SET membership\_name = ?, membership\_duration = ?, membership\_price = ?, membership\_description = ?, membership\_discours = ?, membership\_is\_active = ? WHERE membership\_id = ?

select \* from tblRegistrationMembership rm join tblMembership m on m.membership\_id = rm.membership\_id where rm.account\_id = ?

## 15. View Bill

### a. Class Diagram



### b. Class Specifications

#### Bill Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter | Get value in attributes |
| 02 | Setter | Set value for attributes |
| 03 | Constructor | Generate Bill object |

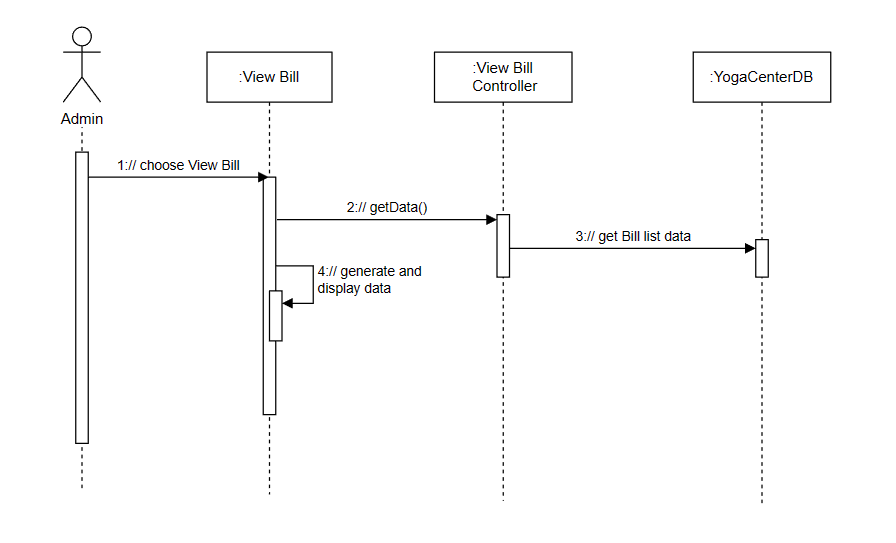
#### Bill Repository

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAllFollowPagination(int offset, int next): | Get all Bill with a range start with “offset” end with “next” |

#### Blog Analysis Controller

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doGet() | Get data from Yoga Center Database, generate and set attributes into request than forward to ViewBillList.jsp |
| 02 | doPost() | Get data from Yoga Center Database, generate and set attributes into request than forward to ViewBillList.jsp |

### c. Sequence Diagram(s)

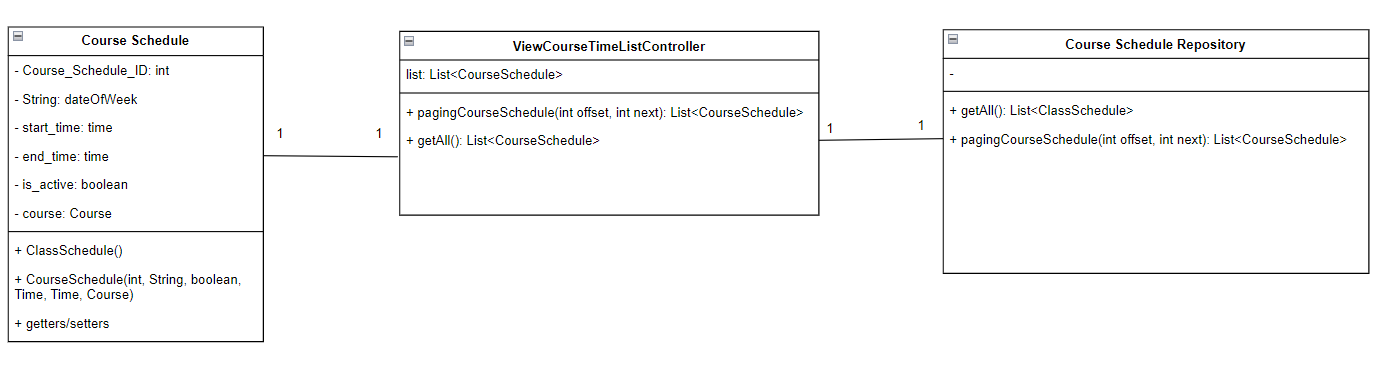


### d. Database Queries

SELECT \* FROM tblBill ORDER BY bill\_id DESC OFFSET ? ROWS FETCH NEXT ? ROWS ONLY

## 16. View Course Time List

### a. Class Diagram



### 

### b. Class Specifications

***Course Schedule Model***

| No | Method | Description |
| --- | --- | --- |
| 01 | Getter/setter | Get value and set value for attributes |
| 02 | Constructor | Generate Course Schedule object |

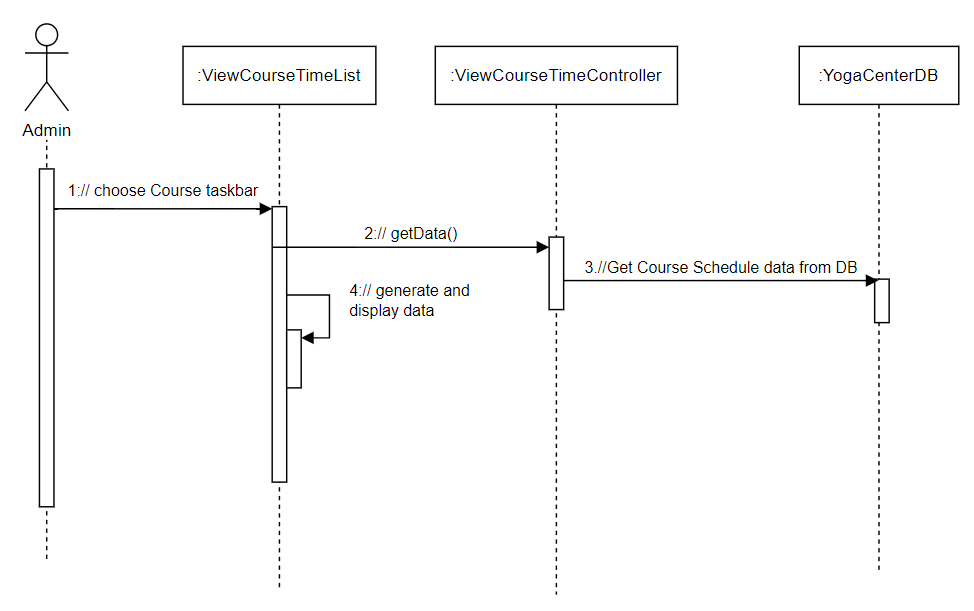
**CourseScheduleRepository Class**

| No | Method | Description |
| --- | --- | --- |
| 01 | pagingCourseSchedule(int offset, int next) | Function used to get a list of data in the database and use OFFSET to paginate this data list |

**ViewCourseTimeListController Class**

| No | Method | Description |
| --- | --- | --- |
| 01 | doget() | Function to getting list data then request dispatch into JSP page |

### c. Sequence Diagram(s)



### 

### d. Database Queries

*SELECT \**

*FROM tblCourseSchedule*

*ORDER BY course\_schedule\_id*

*OFFSET? ROWS FETCH NEXT? ROWS ONLY*

## 17. Cancel Booking

### a. Class Diagram

### 

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

### AccountRepository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll() | Get all accounts in the database where the active field is true, there is no input parameter, the output is the account list |
| 02 | detail() | Get an account in the database where account\_id is equal to the input parameter, the output is an account |
| 03 | update() | Update an account in the database where account\_id is equal to the input parameter, with no output |
| 04 | add() | Add an account into the database, email is not duplicate, input is an account, no output |

#### 

***AccountModel Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | setter() | Set data for attributes in class |
| 02 | getter() | Get data value from attribute in class |
| 03 | AccountModel() | Constructor for class |

#### Bill Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter | Get value in attributes |
| 02 | Setter | Set value for attributes |
| 03 | Constructor | Generate Bill object |

### BillRepository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll() | Get all bills in the database ,there is no input parameter, the output is the account list |
| 02 | detail() | Get an bill in the database where bill\_id is equal to the input parameter, the output is an account |
| 03 | update() | Update an bill in the database where bill\_id is equal to the input parameter, with no output |
| 04 | add() | Add an bill into the database, email is not duplicate, input is an bill, no output |
| 05 | getBillByAccount() | Get all bills in database where account\_id bill equal with account\_id input, output is a Bill list |

#### UserBookingProfileController Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | updateBillStatus() | Update bill status for a bill in database, input is status and bill id, output is boolean |
| 02 | getBillAccount() | Take all bill in database by using BillRepository , input is account id, output is a bill list |
| 03 | printBillList() | Print all bill into UserBookingProfileBill |

#### UserBookingProfileView Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | cancelBill() | Send action to controller and update status bill to cancelled, input is bill id, no output |
| 02 | viewBillDetail() | send request to UserBookingProfileController to get detail for a bill, input is bill id, output is detail of bill |

### 

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

### 

### 

### d. Database Queries

*SELECT \* FROM tblBill*

*SELECT \* FROM tblBill WHERE account\_id=<account\_id>*

*UPDATE tblBill SET bill\_status = ? WHERE bill\_id = <bill\_id>*

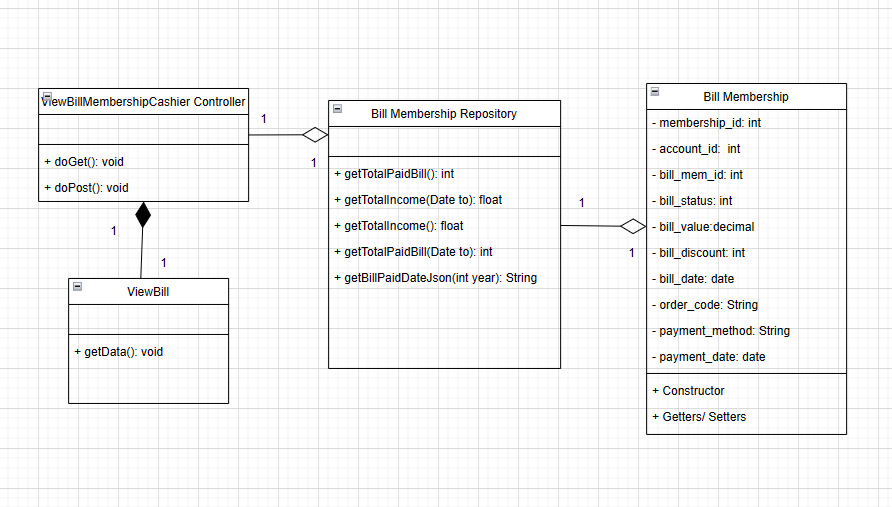
*INSERT INTO tblBill (course\_id, account\_id, bill\_status, bill\_is\_active, bill\_value, bill\_discount, bill\_date, order\_code, payment\_method) VALUES (<course\_id>, <account\_id>, <bill\_status>, <bill\_is\_active>, <bill\_value>, <bill\_discount>, <bill\_date>, <order\_code>, <payment\_method>)*

*SELECT \* FROM tblAccount WHERE account\_id= <account\_id>*

*SELECT \* FROM tblAccount*

## 18. Cashier ViewBillMembership

### a. Class Diagram



### b. Class Specifications

#### Bill Membership Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter | Get value in attributes |
| 02 | Setter | Set value for attributes |
| 03 | Constructor | Generate Bill Mebership object |

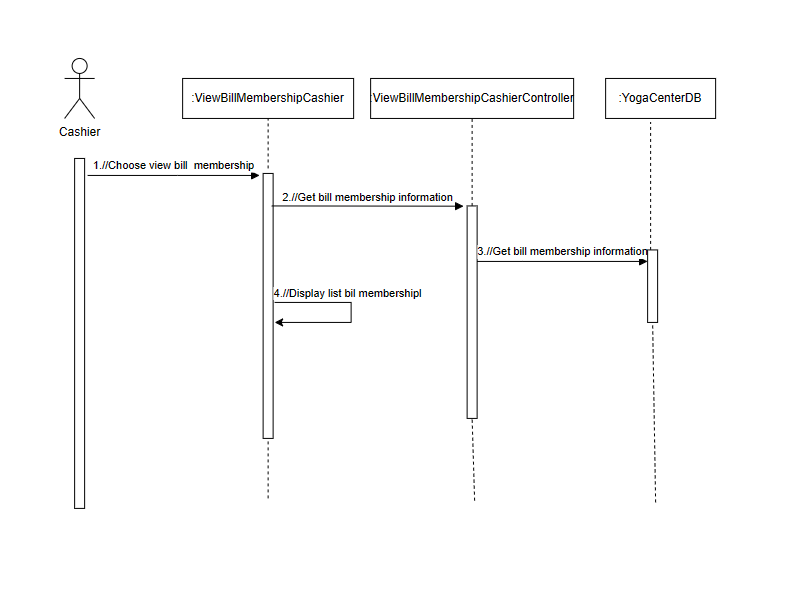
#### Bill Membership Repository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll():List<BillMembership> | Get all bill membership in the database where active is true, the output is a membership list, no input |
| 02 | getByAccountID | Get membership current of an account\_id |
| 03 | getMembershipIdByOrdercode | Get membership current of an order\_code |
| 04 | updateStatus | Update status of bill\_membership\_status where bill\_membership\_id=? |
| 05 | searchOrderCode | Search bill membership by order\_code=? |

#### ViewBillMembershipController Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doGet() | Method get all bill membership by using BillMembershipRepository |

### c. Sequence Diagram(s)

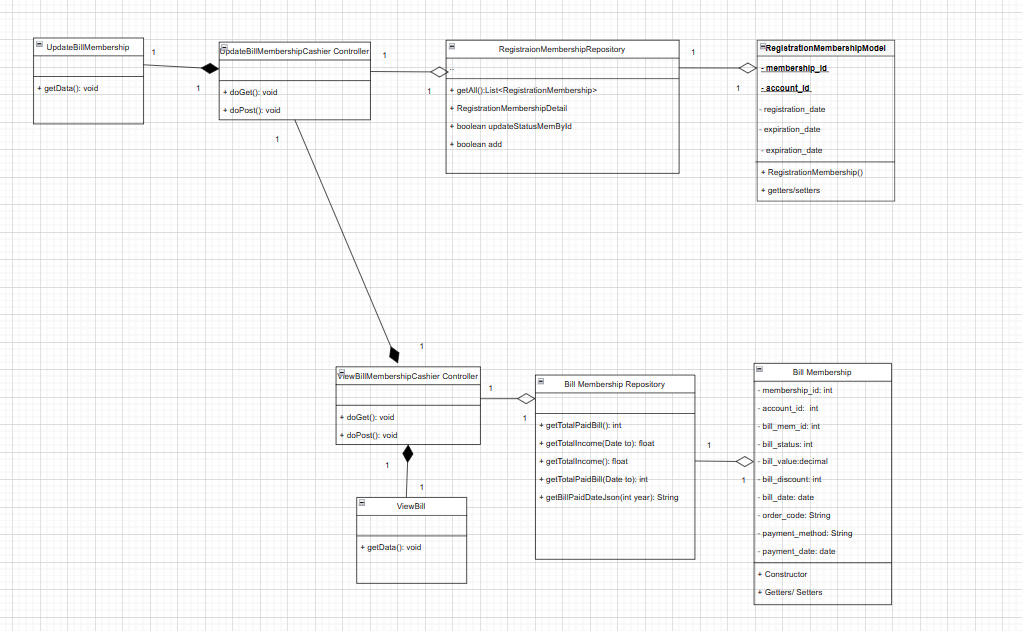


### d. Database Queries

select \* from tblBillMembership

## 19. Cashier UpdateBillMembership

### a. Class Diagram



### b. Class Specifications

#### 

#### Bill Membership Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | Getter | Get value in attributes |
| 02 | Setter | Set value for attributes |
| 03 | Constructor | Generate Bill Membership object |

#### Bill Membership Repository Class

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll():List<BillMembership> | Get all bill membership in the database where active is true, the output is a membership list, no input |
| 02 | getByAccountID | Get membership current of an account\_id |
| 03 | getMembershipIdByOrdercode | Get membership current of an order\_code |
| 04 | updateStatus | Update status of bill\_membership\_status where bill\_membership\_id=? |
| 05 | searchOrderCode | Search bill membership by order\_code=? |

#### 

#### RegistrationMembership Repository Class

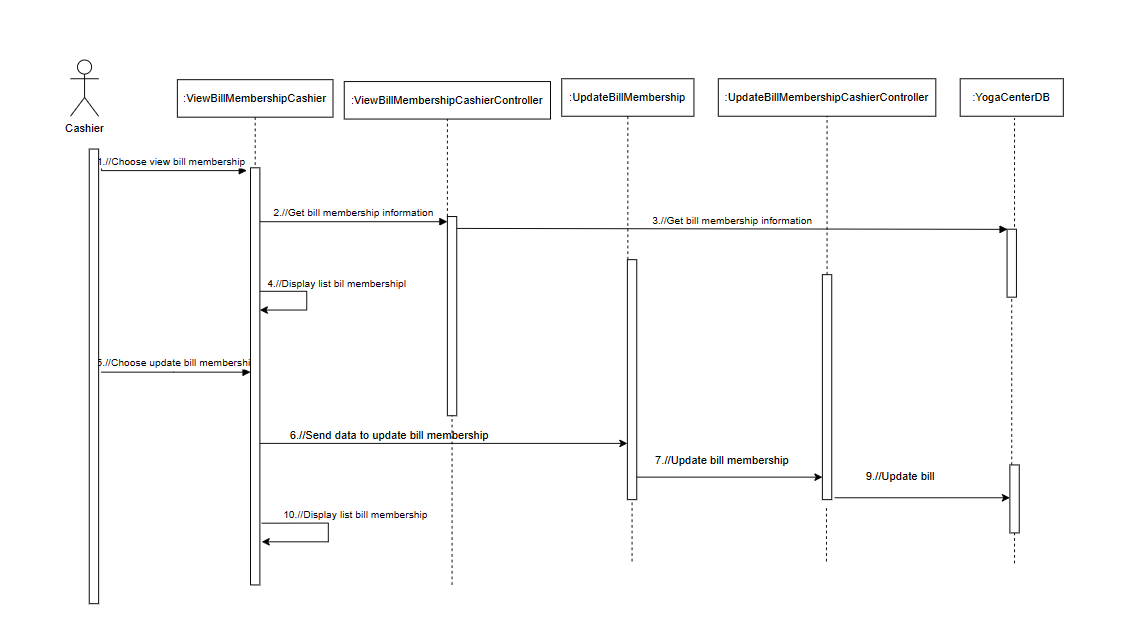
| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | getAll():List<RegistrationMembership> | Get all registration memberships in the database where active is true, the output is a registration membership list, no input |
| 02 | RegistrationMembershipDetail | Get all registration detail in the database by registration id =? |
| 03 | updateStatusMemById | Update status of registration\_status |
| 04 | add | Add new registration membership |

#### 

***UpdateBillMembershipCashierController Class***

| **No** | **Method** | **Description** |
| --- | --- | --- |
| 01 | doGet() | Get information details to update form |
| 02 | doPost() | Update bill\_status and payment\_date and update registration\_status by using BillMembershipRepository and RegistrationMembershipRepository |

### c. Sequence Diagram(s)



### d. Database Queries

select \* from tblBillMembership where bill\_mem\_id=?

UPDATE tblBillMembership SET bill\_status = ? , payment\_date =? WHERE order\_code = ?

select \* from tblBillMembership where order\_code=?

select \* from tblRegistrationMembership WHERE membership\_id = ? AND account\_id = ? order by membership\_id desc

UPDATE tblRegistrationMembership SET registration\_status = ? WHERE membership\_id = ?