

**SWD PROJECT REPORT**

**Software Design Document**

– Hanoi, August 2023 –

**Table of Contents**

[I. Record of Changes 3](#_gjdgxs)

[II. Software Design Document 4](#_30j0zll)

[1. System Design 4](#_1fob9te)

[1.1 System Architecture 4](#_3znysh7)

[1.2 Package Diagram 4](#_2et92p0)

[2. Database Design 4](#_tyjcwt)

[2.1 table\_name1 5](#_3dy6vkm)

[2.2 table\_name2 5](#_1t3h5sf)

[3. Detailed Design 6](#_4d34og8)

[3.1 <Feature/Function Name1> 6](#_2s8eyo1)

[3.2 <Feature/Function Name2> 7](#_17dp8vu)

# I. Record of Changes

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 15/10/2023 | A | DucVD | Add package diagram |
| 15/10/2923 | A | DucVD | Add Database Design |
| 15/10/2023 | A | DucVD | Add Package Description |
| 21/10/2023 | A | DucVD | Add table user information |
| 21/10/2023 | A | DucVD | Add table project information |
| 21/10/2023 | A | PhongVD | Add table milestone information |
| 21/10/2023 | A | PhongVD | Add table issues information |
| 21/10/2023 | A | PhongVD | Add table classes information |
| 21/10/2023 | A | PhongVD | Add register sequence diagram |
| 21/10/2023 | A | DucVD | Add login sequence diagram |
| 24/10/2023 | A | TruongNQ | System Architecture |
| 24/10/2023 | M | DungND | System Architecture |
| 25/10/2023 | M | TruongNQ | Database Design |
| 5/11/2023 | A | TruongNQ | Class Diagram |
| 5/11/2023 | A | TruongNQ | Subject Management |
| 5/11/2023 | A | TruongNQ | Project Management |
| 5/11/2023 | A | TruongNQ | Issue Management |
| 5/11/2023 | A | TruongNQ | Class Management |
| 5/11/2023 | A | TruongNQ | Milestone Management |
| 5/11/2023 | A | TruongNQ | System Setting |
| 5/11/2023 | A | TruongNQ | Class Specifications |

\*A - Added M - Modified D - Deleted

# II. Software Design Document

## 1. System Design

### 1.1 System Architecture

● MVC(Model - View - Controller)  
Model: Represents the data and logic of the application. It handles data-related operations, including retrieving, updating, and handling business logic.

View: Represents the user interface, displays data to the user and interacts with the user. It represents data from the Model and displays it to the user in an appropriate format.

Controller: Is the application's controller, handling events from users and controlling data flow between Model and View. It receives commands from the user through the View interface, then interacts with the Model to update or retrieve data and updates the View interface to display the results.

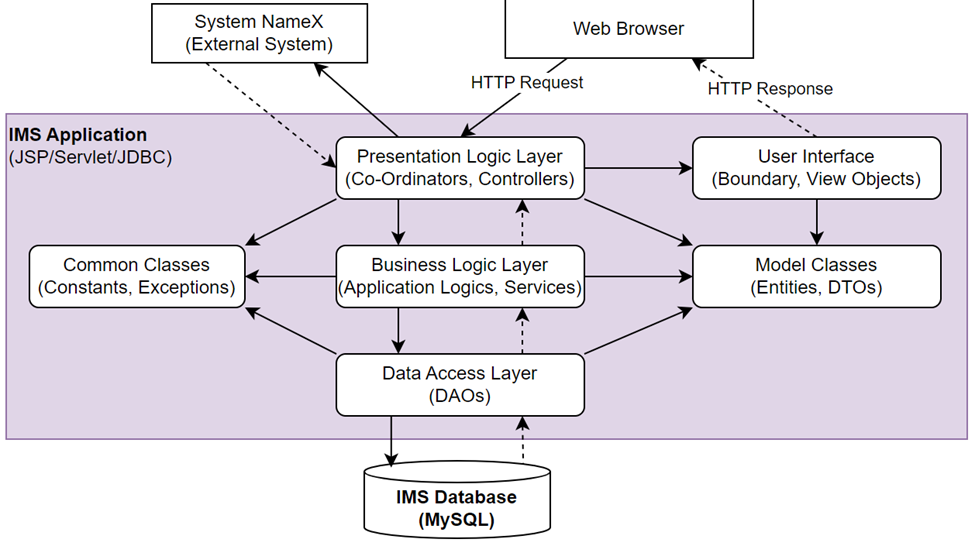
● Java Web Servlet

● The app uses servlet to listen to requests from front-end, front-end uses HTML, CSS, JS in JSP file

● Database using SQL Server.

● Web Application: Tomcat

● Server: Servlet



| **No** | **Component** | **Description** |
| --- | --- | --- |
| 01 | Web Browser | The web interface that users use to interact with the application |
| 02 | User Interface | The user interface part of the application, including components such as buttons, forms, and interactive elements. |
| 03 | Presentation Logic Layer | Handles logic related to data display and user interaction. |
| 04 | Business Logic Layer | Handles business logic and manages application workflow. |
| 05 | Data Access Layer | Link to the database and perform data read and write operations |
| 06 | Common Classes | Contains common classes and components used throughout the application. |
| 07 | System | This is the layer that contains system configuration information or common components of the entire application. |

### 1.2 Package Diagram

*[Provide the package diagram for each sub-system. The content of this section including the overall package diagram, the explanation, package and class naming conventions in each package. Please see the sample & description table format below]*

Ảnh có chứa văn bản, bầu trời, trắng

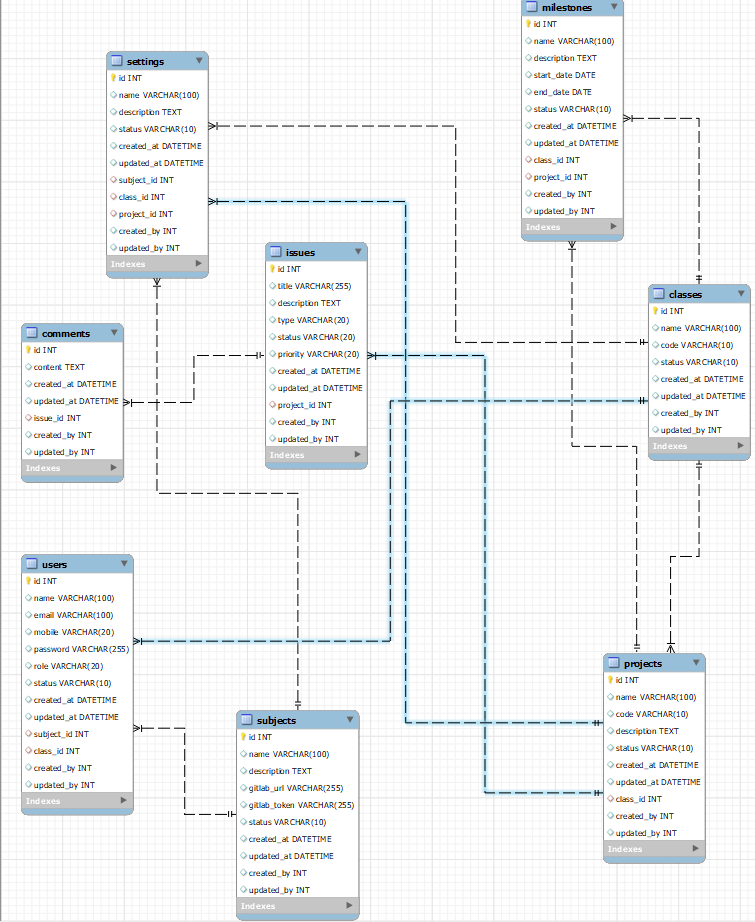
Mô tả được tạo tự động

***Package descriptions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *Account* | *Provide all function login log out, captcha,...* |
| *02* | *Model* | *Model of account* |
| *03* | *Tool* | *Tool use in log in log out function* |
| *04* | *DAO* | *Provide SQL statement and connect to database* |

## 2. Database Design

*[Provide the files description, database table relationship & table descriptions like example below]*



### 

### 

### 

### 2.1 User

Provide all the profile of user

[Provide the detailed table fields description using below table format  
 \* PK~Primary Key; FK~Foreign Key; UN~Unique; NN ~ not null

| **No** | **Field** | **PK** | **FK** | **UN** | **NN** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| *01* | *id* | *x* |  | *x* | *x* | *identity key* |
| *02* | *name* |  |  |  | *x* | *Name of user* |
| *03* | *email* |  |  |  | *x* | *User email* |
| *04* | *mobile* |  |  |  | *x* | *phone number* |
| *05* | *password* |  |  |  | *x* | *Password* |
| *06* | *role* |  |  | *x* | *x* | *role of user* |
| *07* | *status* |  |  |  | *x* | *user status* |
| *08* | *created\_at* |  |  | *x* | *x* | *define where it create* |
| *09* | *update\_at* |  |  | *x* | *x* | *define where it update* |
| *10* | *subject\_id* |  | *x* |  | *x* | *ID of subject* |
| *11* | *class\_id* |  | *x* |  | *x* | *ID of Class* |
| *12* | *create\_by* |  |  | *x* | *x* | *define who create* |
| *13* | *update\_by* |  |  | *x* | *x* | *define who update* |

### 2.2 subject

Provide the subject information

| **No** | **Field** | **PK** | **FK** | **UN** | **NN** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| *01* | *id* | *x* |  | *x* | *x* | *identity key* |
| *02* | *name* |  |  |  | *x* | *Name of subject* |
| *03* | *description* |  |  |  | *x* | *Subject description* |
| *04* | *gitlab\_url* |  |  | *x* | *x* | *gitlab URL* |
| *05* | *gitlab\_token* |  |  | *x* | *x* | *Gitlab token* |
| *06* | *status* |  |  |  | *x* | *Subject status* |
| *07* | *created\_at* |  |  | *x* | *x* | *subject create at* |
| *08* | *update\_at* |  |  | *x* | *x* | *subject update at* |
| *09* | *create\_by* |  |  | *x* | *x* | *subject create by* |
| *10* | *update\_by* |  |  | *x* | *x* | *subject update by* |

### 

### 2.3 project

Provide the project information

| **No** | **Field** | **PK** | **FK** | **UN** | **NN** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| *01* | *id* | *x* |  | *x* | *x* | *identity key* |
| *02* | *name* |  |  |  | *x* | *Project Name* |
| *03* | *description* |  |  |  | *x* | *Project description* |
| *04* | *gitlab\_url* |  |  |  | *x* | *Gitlab URL* |
| *05* | *gitlab\_token* |  |  |  | *x* | *Gitlab token* |
| *06* | *status* |  |  |  | *x* | *project status* |
| *07* | *created\_at* |  |  | *x* | *x* | *project create at* |
| *08* | *update\_at* |  |  | *x* | *x* | *project update at* |
| *09* | *create\_by* |  |  | *x* | *x* | *project create by* |
| *10* | *update\_by* |  |  | *x* | *x* | *project update by* |
| *11* | *code* |  |  |  | *x* | *project code* |
| *12* | *class\_id* |  | *x* |  | *x* | *ID of class* |

### 2.4 classes

| **No** | **Field** | **PK** | **FK** | **UN** | **NN** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| *01* | *id* | *x* |  | *x* | *x* | *identity key* |
| *02* | *name* |  |  |  | *x* | *Class Name* |
| *03* | *created\_at* |  |  | *x* | *x* | *Class create at* |
| *04* | *update\_at* |  |  | *x* | *x* | *Class update at* |
| *05* | *create\_by* |  |  | *x* | *x* | *Class create by* |
| *06* | *update\_by* |  |  | *x* | *x* | *Class update by* |
| *07* | *code* |  |  |  | *x* | *Class code* |

### 2.5 issues

| **No** | **Field** | **PK** | **FK** | **UN** | **NN** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| *01* | *id* | *x* |  | *x* | *x* | *identity key* |
| *02* | *priority* |  |  |  | *x* | *Issue priority* |
| *03* | *description* |  |  |  | *x* | *Issue description* |
| *04* | *title* |  |  |  | *x* | *Issue title* |
| *05* | *type* |  |  |  | *x* | *Issue type* |
| *06* | *status* |  |  |  | *x* | *Issue status* |
| *07* | *created\_at* |  |  | *x* | *x* | *Issue create at* |
| *08* | *update\_at* |  |  | *x* | *x* | *Issue update at* |
| *09* | *create\_by* |  |  | *x* | *x* | *Issue create by* |
| *10* | *update\_by* |  |  | *x* | *x* | *Issue update by* |
| *11* | *code* |  |  |  | *x* | *Issue Code* |
| *12* | *projecy\_id* |  | *x* |  | *x* | *ID of project* |

### 2.6 milstone

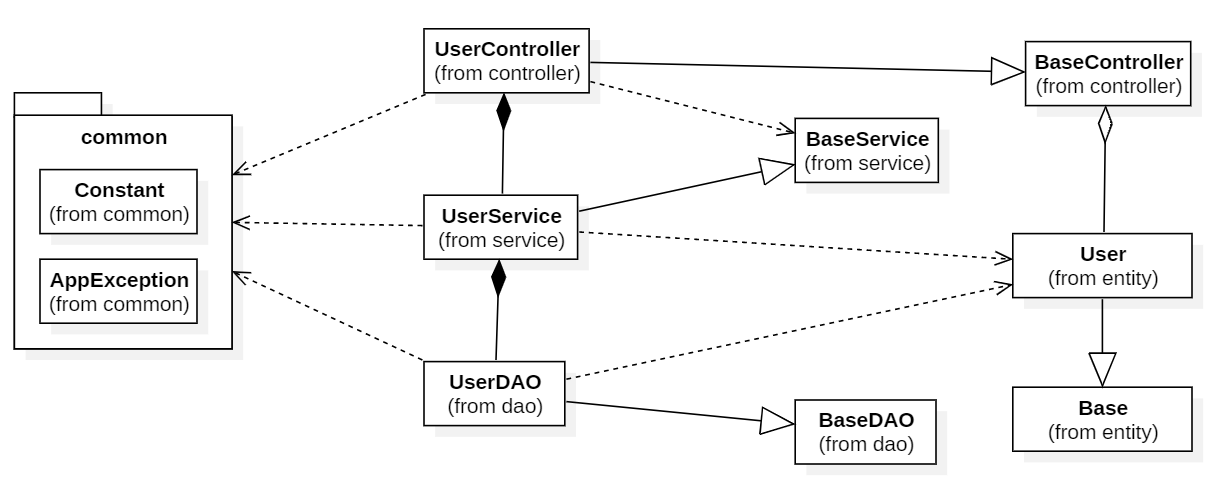
| **No** | **Field** | **PK** | **FK** | **UN** | **NN** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| *01* | *id* | *x* |  | *x* | *x* | *identity key* |
| *02* | *name* |  |  |  | *x* | *milestone Name* |
| *03* | *description* |  |  |  | *x* | *milestone description* |
| *04* | *start\_date* |  |  | *x* | *x* | *Date start* |
| *05* | *end\_date* |  |  | *x* | *x* | *Date end* |
| *06* | *status* |  |  |  | *x* | *milestone status* |
| *07* | *created\_at* |  |  | *x* | *x* | *milestone create at* |
| *08* | *update\_at* |  |  | *x* | *x* | *milestone update at* |
| *09* | *create\_by* |  |  | *x* | *x* | *milestone create by* |
| *10* | *update\_by* |  |  | *x* | *x* | *milestone update by* |
| *11* | *project\_id* |  | *x* |  | *x* | *ID of project* |
| *12* | *class\_id* |  | *x* |  | *x* | *ID of class* |

## 3. Detailed Design

### 3.1 User Management

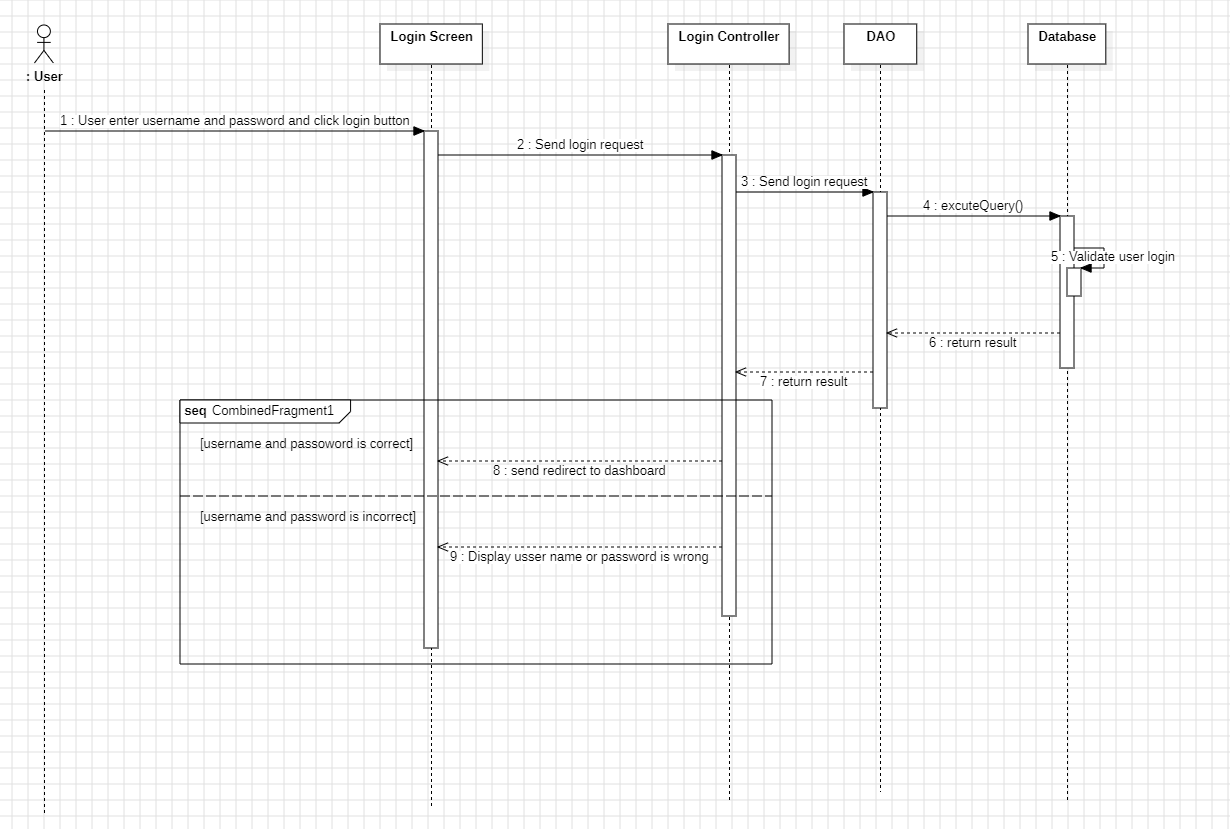
#### 3.1.1 Class Diagram

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

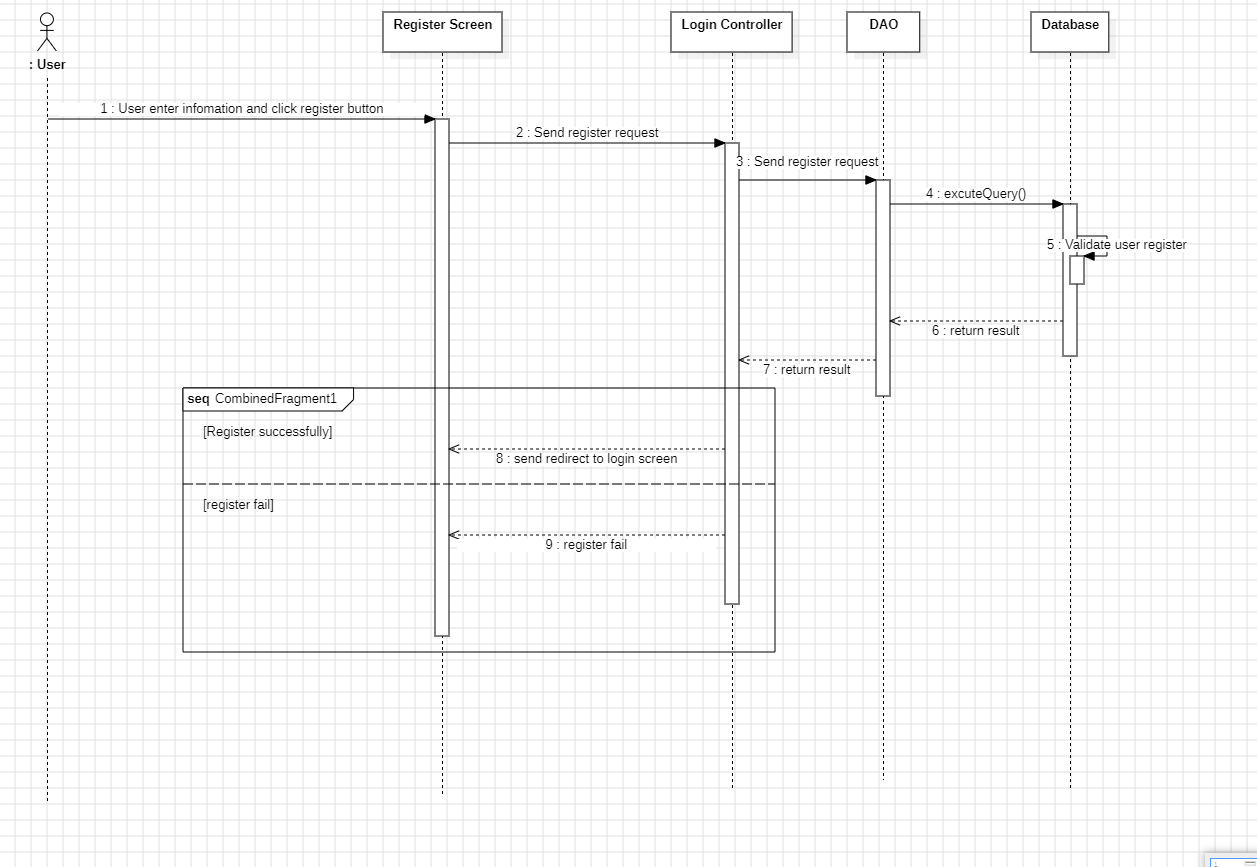


#### 3.1.2 Login

*[Provide the sequence diagram(s) for the feature, see the sample below]*

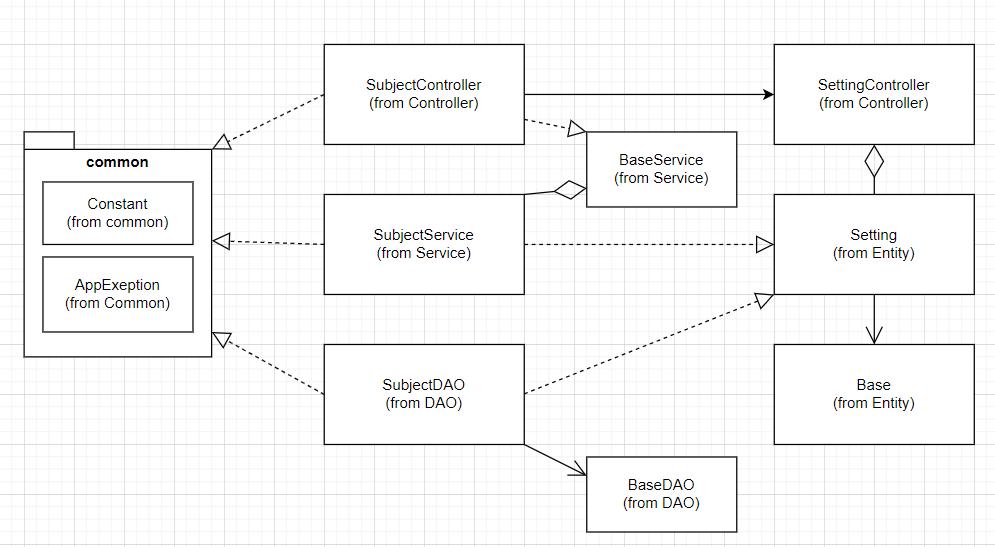


#### 3.1.3 Register

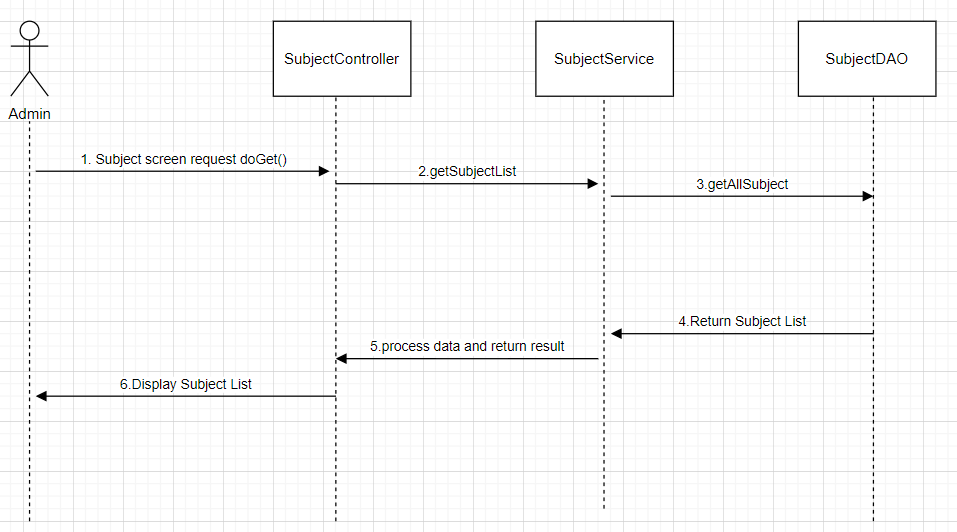


### 3.2 Subject Management

**3.2.1. Class diagram**

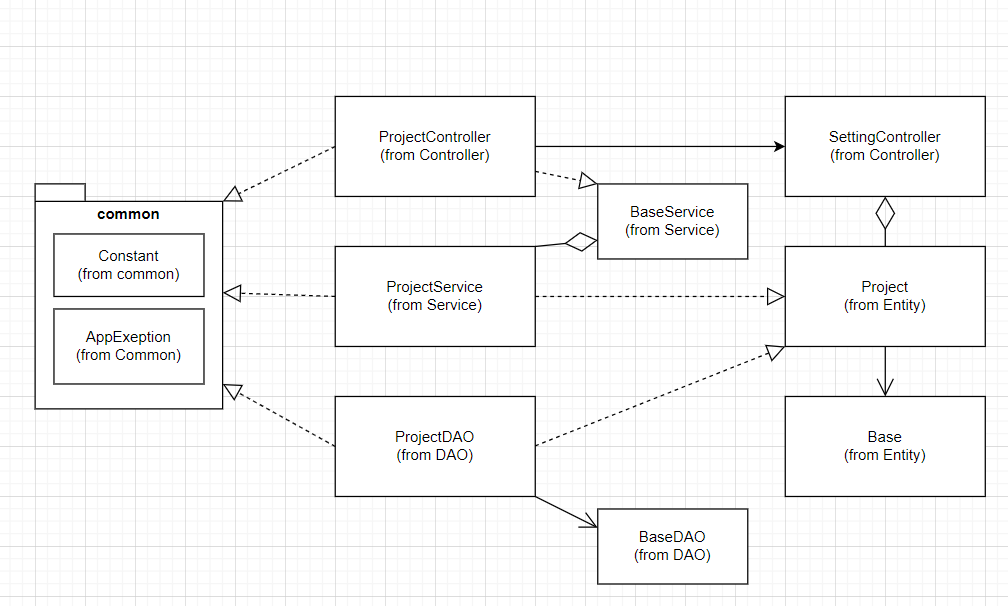


3.3.2.Subject List



### 3.3 Project Management

**3.3.1. Class diagram**

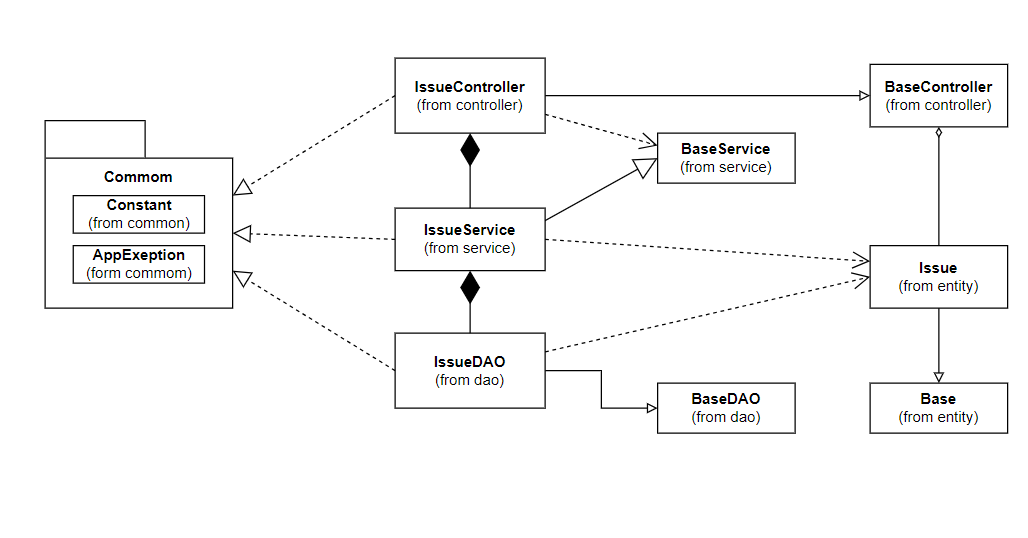


**3.3.2. View project**

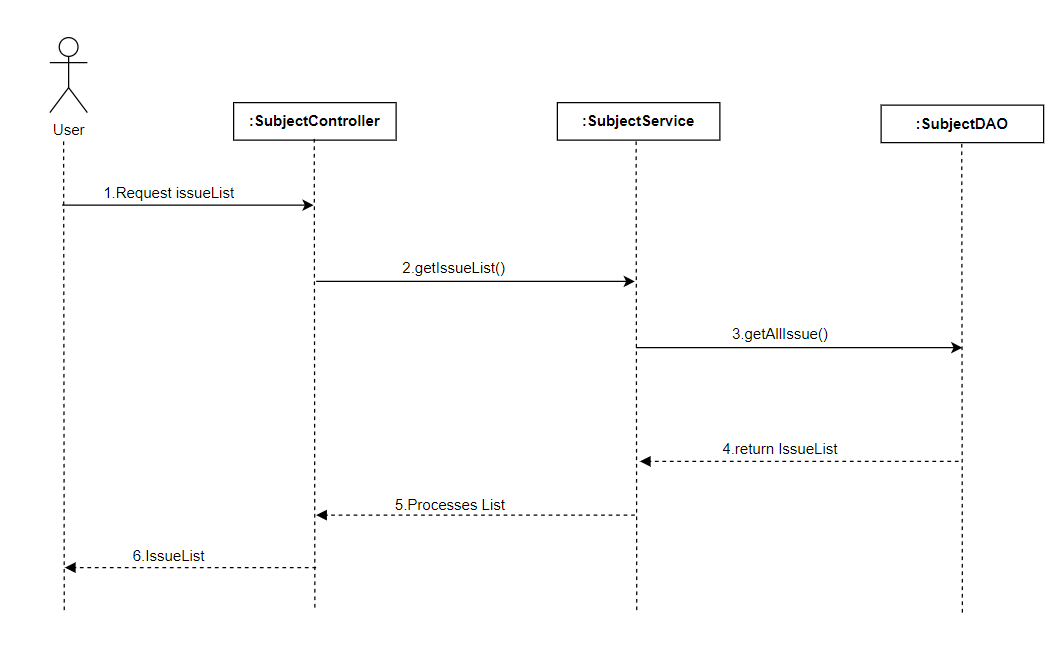
#### 

### 3.4 Issue Management

3.4.1 Class diagram

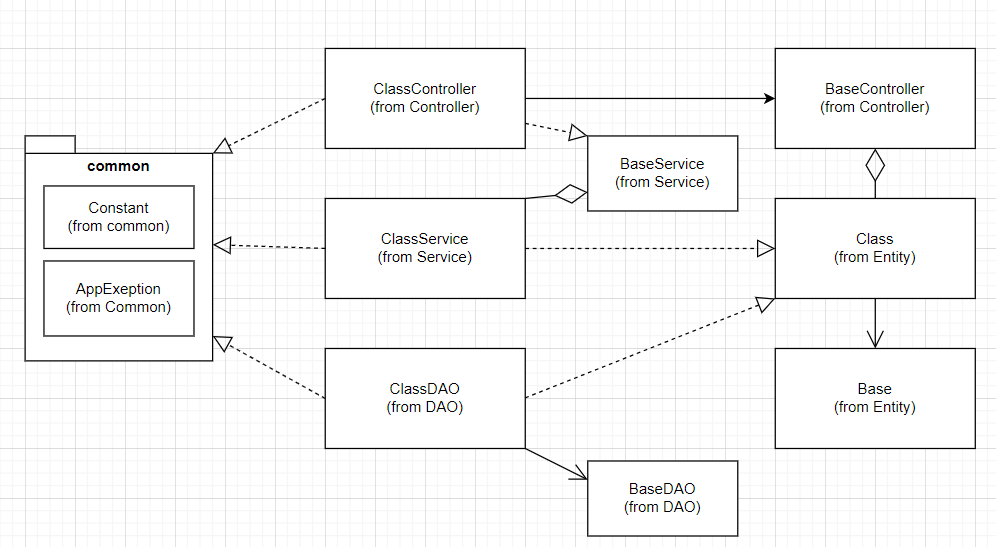


3.4.2. List Issue setting

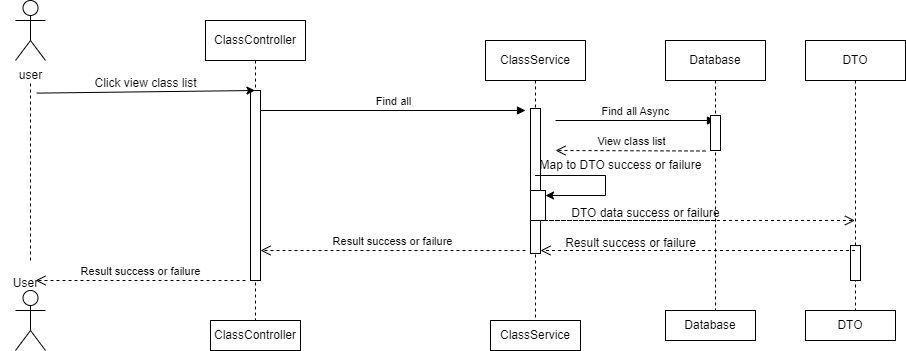


### 3.5 Class Management

3.5.1 Class diagram



3.5.2. View Class List

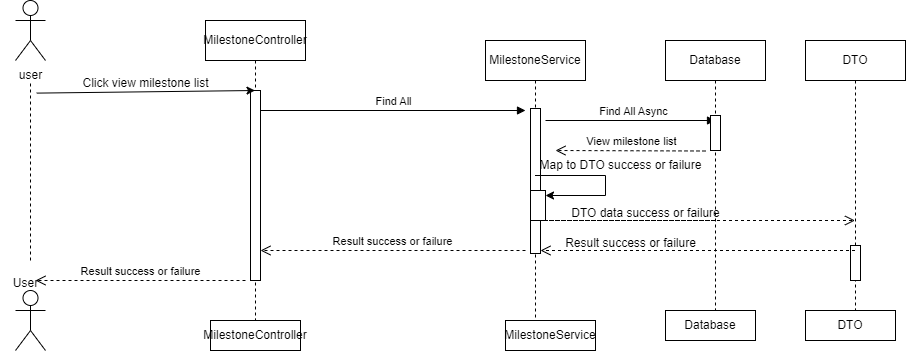


### 

### 3.6 Milestone Management

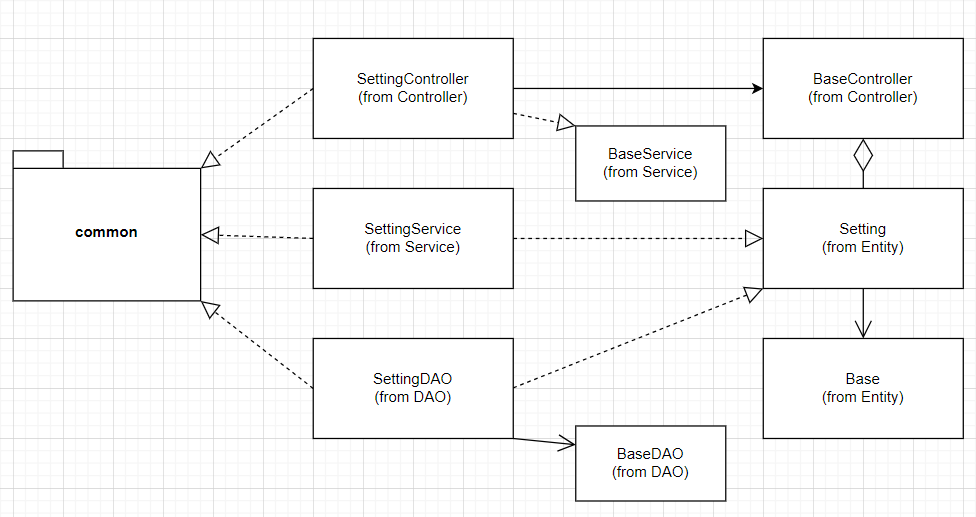
3.6.1 Class diagram

**3.6.2. View Milestone list**



### 3.7 System Setting

3.7.1 Class diagram



## 4. Class Specifications

### 4.1 Model

#### 4.1.1 Issue

| **No** | **Name** | **Description** |
| --- | --- | --- |
| ***Attributes*** | | |
| *01* | *Add Issue* | Visibility: private  Type:String  Purpose:Add issue into list |
| *02* | Delete Issue | Visibility: private  Type:String  Purpose:Delete Issue |
| *03* | Issue Detail | Visibility: private  Type:String  Purpose:*Show Issue Detail* |
| *04* | Update Issue | Visibility: private  Type:String  Purpose:*Update Issue* |
| *05* | Issue List | Visibility: private  Type:String  Purpose:*List Issue* |
| ***Methods/Operations*** | | |
| *01* | *toString* | Visibility:private  Return:*String*  Purpose: *return the value of item* |
| *02* | main | Visibility:private  Return:*Static*  Purpose: *display the value* |

#### 4.1.2 Class

| **No** | **Name** | **Description** |
| --- | --- | --- |
| ***Attributes*** | | |
| *01* | *Add Class* | Visibility: private  Type:String  Purpose:Add Class into list |
| *02* | Delete Class | Visibility: private  Type:String  Purpose:Delete Class |
| *03* | ClassDetail | Visibility: private  Type:String  Purpose:*Show* Class *Detail* |
| *04* | Update Class | Visibility: private  Type:String  Purpose:*Update* Class |
| *05* | Class List | Visibility: private  Type:String  Purpose:*List* Class |
| ***Methods/Operations*** | | |
| *01* | *toString* | Visibility:private  Return:*String*  Purpose: *return the value of item* |
| *02* | main | Visibility:private  Return:*Static*  Purpose: *display the value* |

#### 4.1.3 Setting

| **No** | **Name** | **Description** |
| --- | --- | --- |
| ***Attributes*** | | |
| *01* | *Add Setting* | Visibility: private  Type:String  Purpose:Add *Setting* into list |
| *02* | Delete *Setting* | Visibility: private  Type:String  Purpose:Delete *Setting* |
| *03* | *Setting*Detail | Visibility: private  Type:String  Purpose:*Show SettingDetail* |
| *04* | Update *Setting* | Visibility: private  Type:String  Purpose:*Update Setting* |
| *05* | *Setting*List | Visibility: private  Type:String  Purpose:*List Setting* |
| ***Methods/Operations*** | | |
| *01* | *toString* | Visibility:private  Return:*String*  Purpose: *return the value of item* |
| *02* | main | Visibility:private  Return:*Static*  Purpose: *display the value* |

#### 4.1.4 Subject

| **No** | **Name** | **Description** |
| --- | --- | --- |
| ***Attributes*** | | |
| *01* | *Add Subject* | Visibility: private  Type:String  Purpose:Add *Subject into* list |
| *02* | Delete *Subject* | Visibility: private  Type:String  Purpose:Delete *Subject* |
| *03* | *Subject*Detail | Visibility: private  Type:String  Purpose:*Show Subject Detail* |
| *04* | Update *Subject* | Visibility: private  Type:String  Purpose:*Update Subject* |
| *05* | *Subject List* | Visibility: private  Type:String  Purpose:*List Subject* |
| ***Methods/Operations*** | | |
| *01* | *toString* | Visibility:private  Return:*String*  Purpose: *return the value of item* |
| *02* | main | Visibility:private  Return:*Static*  Purpose: *display the value* |