

**Online learning Application**

**Software Design Document**

– Hanoi, May 2022 –

**Table of Contents**

[I. Overview 3](#_Toc69802609)

[1. Code Packages 3](#_Toc69802610)

[2. Database Schema 3](#_Toc69802611)

[II. Code Designs 4](#_Toc69802612)

[1. <Feature/Function Name1> 4](#_Toc69802613)

[a. Class Diagram 4](#_Toc69802614)

[b. Class Specifications 4](#_Toc69802615)

[c. Sequence Diagram(s) 4](#_Toc69802616)

[d. Database queries 5](#_Toc69802617)

[2. <Feature/Function Name2> 5](#_Toc69802618)

[III. Database Tables 5](#_Toc69802619)

[1. <Table name 1> 5](#_Toc69802620)

[2. <Table name 2…> 5](#_Toc69802621)

# I. Overview

## 1. Code Packages/Namespaces

*[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 and description table format below – following Java project naming convention]*



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

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *1* | model | *<Description of the package>*  *<Class naming convention>* |
| *2* | view |  |
| *3* | *controller* |  |
| *4* | *dao* |  |
| *5* | *service* |  |
| *6* | *util* |  |
| *7* | *filter* | *Includes classes that implement the application's Filter interface. These classes will receive and process requests before this request reaches the classes in the controller. The classes in the package implement authorization functions to restrict access to restricted resources and perform unauthorized functions.* |

## 2. Coding Conventions

// Nho dien vao day

# II. Code Designs

## 1. <Feature/Function Name1>

*[Provide the detailed design for the function <Feature/Function Name1>. It include Class Diagram, Class Specifications, and Sequence Diagram(s)]*

### a. Class Diagram

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



### b. Class Specifications

#### DashboardFilter Class

#### Block requests to access the Dashboard screen based on account permissions. Only accounts with Sales, Marketing and Admin permissions have access to the Dashboard when making a request.

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doFilter()* | *Block requests to access the Dashboard, requests when insufficient permissions will be redirected to a message page with content that cannot be accessed because of insufficient permissions.* |

**MyCourseFilter Class**

*When request to query data about registered courses if not logged in or already logged in but this request with intent to get course data of another account then request will fail.*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doFilter()* | *Making a request to get data about registered courses when not logged in will redirect to the login screen.*  *Make a request to get data about registered courses when logged in, but this request includes parameters that are geared towards retrieving data with the intention of retrieving information other than those of the currently logged in account. request will not be fulfilled and redirect to error page.* |

#### PostManagerFilter Class

*Requests to list posts, edit posts, delete posts based on accounts with insufficient permissions will fail. Only accounts with Marketing, Admin rights can access.*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doFilter()* | *Requests to enter the post edit screen, list posts, delete posts without permission will be redirected to the 'Error Page'* |

#### SliderManagerFilter Class

*Requests to list slider, edit slider, delete slider based on accounts with insufficient permissions will fail. Only accounts with Marketing, Admin rights can access.*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doFilter()* | *Requests to enter the slider edit screen, list sliders, delete sliders without permission will be redirected to the 'Error page'* |

#### LessonContentFilter Class

*Handle requests for access to any course content.*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doFilter()* | *When not logged in if a request is made, the user is redirected to the 'Introduce Course' page.*  *When logged in but accessing course content that the user is not registered with, the action will be redirected to the 'Introduce Course' page.*  *With an account that has registered for the course and has Admin rights, access will be successful.* |

#### AccountManagerFilter Class

*All accounts with non-Admin permissions when performing access to the account management screen will not be allowed.*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doFilter()* | *Requests to enter the account edit screen, list accounts, edit accounts that do not have permission will be redirected to the ‘Error Page'.* |

#### AccountSettingFilter Class

*Performing access to the account's configuration page when the account's login session does not exist is not performed.*

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doFilter()* | *Redirects to the login screen when trying to access the account's configuration page when no account login session exists.* |

#### CourseManagerFilter Class

Block and redirect requests to edit, list, and manage courses. Only accounts with Expert and Admin permissions can access the above screens.

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *doFilter()* | *When not logged in will redirect to the login screen.*  *Using an account with unauthorized permissions will redirect to an error page.*  *When using an account with Expert privileges, feedback and changes will be limited depending on the resource that belongs to that Expert only. With Admin rights, resources will not be limited.* |

### c. Sequence Diagram(s)

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



### d. Database queries

*[Provide the detailed SQL (select, insert, update...) which are used in implementing the function/screen]*

## 2. <Feature/Function Name2>

…

# 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* |  |  |