UNIVERSITY OF BARISHAL



Assignment

Course Title : Software Engineering and Information System Design

Course Code : CSE-3103

Topic : Your Eligible Subjects Checker for BU after Admission

Submitted to

Md. Erfan

Assistant Professor,

Department of Computer Science and Engineering,

University of Barishal.

Submitted by

Name : Protap Mistry

Roll : 17CSE029

Session : 2016-2017

Semester : First

Year : Third

Department: Computer Science and Engineering

Date of Submission: 22-10-2021

Page number: 1

**Problem Statement**

A checker system of an examinee to see his/her eligible subject(s) for University of Barishal after admission. It is very difficult for us to implement too many conditions for all subjects. Design patterns solve this problem easily. In Chain of Responsibility design pattern, we can check every subject whether it is eligible or not and check next chain. In this way, this problem can be solved.

**What, Why and How**

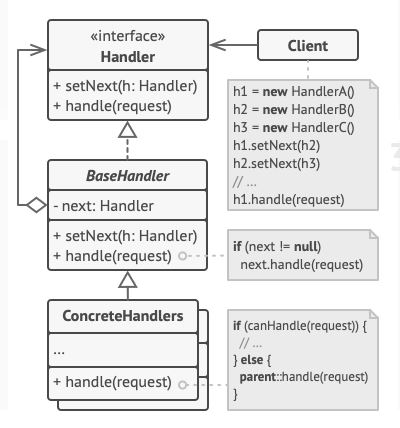
What:

**Chain of Responsibility** is a behavioral design pattern that lets you pass requests along a chain of handlers. Upon receiving a request, each handler decides either to process the request or to pass it to the next handler in the chain.

Why:

I used chain of responsibility design pattern for my application because it’s

1. Reduce the coupling degree. Decoupling it will request the sender and receiver.
2. Simplified object. The object does not need to know the chain structure.
3. Enhance flexibility of object assigned duties. By changing the members within the chain or change their order, allow dynamic adding or deleting responsibility.
4. Increase the request processing new class of very convenient.

How: 

Page number: 2

1. The Handler declares the interface, common for all concrete handlers. It usually contains just a single method for handling requests, but sometimes it may also have another method for setting the next handler on the chain.
2. The Base Handler is an optional class where you can put the boilerplate code that’s common to all handler classes.

Usually, this class defines a field for storing a reference to the next handler. The clients can build a chain by passing a handler to the constructor or setter of the previous handler. The class may also implement the default handling behavior: it can pass execution to the next handler after checking for its existence.

**Solution Strategy**

1. According to CoR design pattern, declare an interface class and describe the methods for handling request and also decide how the client will pass the request data into the method.
2. Creating an abstract base class, derived from the interface class. This class should have a field for storing a reference to the next handler in the chain.
3. Now create concrete handler subclasses and implement their handling methods for each subjects.
4. After that, create instances for every subjects and make a chain with method in the controller (as main class) and pass the request to the chain. This will integrate eligible subjects in the array list.
5. When hit the API end-point route then get valid data in JSON format.

**Implementation**

1. First of all, I create a fresh Laravel project.
2. Create a controller as a main class.
3. Then declare the route and URL with POST method into the API of Laravel.
4. Create a function into the controller that mentioned into the API file.
5. Create interface class for declaring the methods for handling requests and abstract based class for chaining subclasses.
6. Then create subclasses for every subjects which will process the request. Every subject class will check whether it is eligible or not. If it is eligible, then it will store the name of the subject in an array. As this is call by reference, so we should take the input from the subclasses.
7. Declare every class for the design pattern in controller.
8. Validate for all exceptions into the function.
9. Then create instances for every subclass to get eligible subject.
10. Create a chain with the method declared in abstract based class and interface. Now send the request which is in the first chain. Every class will check whether this subject is eligible or not and store it in if eligible and send request to the next chain.
11. In this way an array of the eligible subject list will be implemented and send this array output as JSON format as API response.

Page number: 3

1. With POST method, test with Postman software using API end-point URL and get

/\*\* Input format \*\*/

{

    "hsc\_mathematics\_subj": 0,

    "hsc\_biology\_subj": 1,

    "admission\_mathematics\_marks": 9,

    "admission\_physics\_marks": 5,

    "admission\_chemistry\_marks": 2,

    "admission\_biology\_marks": 6

}

In order to checking availability for H.S.C. subjects take datatype Boolean (0/1) as input and min marks be -2.5 and max marks be 10 for each subject.

/\*\* Sample Output\*\*/

{

    "message": "Your eligible subject(s) for 'A' unit of University of Barishal.",

    "subj": [

        "Computer Science and Engineering",

        "Botany",

        "Coastal Studies and Disaster Management",

        "Bio-chemistry and Bio-technology"

    ]

}